Campus de l'Université de Montréal
C.P. 6128, succursale Centre-ville
Montréal QC H3C 3J7 Canada

Telephone (514) 343-7575
Telecopieur (514) 343-7121

crt@crt.umontreal.ca
www.crt.umontreal.ca

CRT-2001-46

Centre de recherche sur les transports (C.R.T.)
Centre for Research on Transportation

PROBLÈMES DE GESTION DE FLOTTES DE VÉHICULES EN TEMPS RÉEL

par

Soumia Ichoua

THÈSE DE DOCTORAT

Novembre 2001

PUBLICATION
CRT-2001-46

PROBLÈMES DE GESTION DE FLOTTES DE VÉHICULES EN TEMPS RÉEL*

par

Soumia Ichoua[1]

* Thèse présentée en vue de l'obtention du grade de Docteure ès sciences (Ph.D.) en informatique (Directeur de recherche : Michel Gendreau; codirecteur de recherche : Jean-Yves Potvin).

[1] Département d'informatique et de recherche opérationnelle et Centre de recherche sur les transports, Université de Montréal, C.P. 6128, succursale Centre-ville, Montréal, Canada H3C 3J7

©Soumia Ichoua, 2001

Dépôt légal – Bibliothèque nationale du Québec
Bibliothèque nationale du Canada, 2001

Sommaire

Cette thèse porte sur le problème de répartition d'une flotte de véhicules en temps réel. Dans ces problèmes, il s'agit d'affecter des requêtes qui arrivent en temps réel à une flotte de véhicules en mouvement. Ce processus implique aussi la construction simultanée d'un ensemble de routes planifiées qui satisfont la demande, tout en respectant certaines contraintes.

Que l'on soit intéressé par le transport de biens ou de personnes, un grand défi à l'heure actuelle est d'élaborer des approches de résolution rapides mais efficaces dans un environnement qui évolue dans le temps. Notre thèse s'oriente vers ce nouvel axe de recherche qui est en pleine expansion grâce aux progrès technologiques récents (p.e. positionnement par satellite, télé-détection, etc...). Nous nous attardons sur trois problématiques rencontrées dans les problèmes de gestion de flotte en temps réel et nous proposons des stratégies de résolution pour ces problématiques.

Les principales contributions de cette thèse se résument comme suit. Dans un premier temps, nous présentons une revue et une synthèse de la littérature consacrée aux problèmes de répartition de véhicules en temps réel. Nous dressons un inventaire non exhaustif des contextes d'application les plus répandus. Par la suite, nous classifions ces différents groupes de problèmes en nous basant sur une taxinomie établie au préalable. Nous discutons également des méthodologies générales adoptées pour traiter l'aspect dynamique de ces problèmes et nous dégageons quelques avenues de recherche qu'il serait pertinent d'approfondir dans le cadre de travaux futurs.

Dans le premier article, nous présentons une vision plus large du concept de la

diversion. Nous examinons également différentes façons d'intégrer ce concept à des algorithmes qui ne sont pas initialement conçus pour en tenir compte. Finalement, nous évaluons les bénéfices de son utilisation.

Dans le second article, nous proposons et analysons un modèle basé sur des temps de parcours variables dans le temps qui s'applique aussi bien dans un contexte statique que dans un contexte dynamique. Ensuite, nous comparons le modèle proposé à un modèle où les temps de parcours sont constants.

Dans le troisième article, nous proposons une stratégie qui exploite une anticipation des demandes futures afin d'améliorer la qualité des solutions obtenues. En particulier, la stratégie proposée introduit des demandes fictives (représentant des requêtes anticipées) dans les routes des véhicules afin d'assurer une bonne couverture territoriale. Nous évaluons enfin les bénéfices d'une telle stratégie pour différents scénarios d'opérations.

Les différentes problématiques étudiées sont abordées en incorporant les stratégies proposées au sein de méthodes heuristiques basées sur la recherche avec tabous. Afin de faire face aux pressions temporelles inhérentes à un contexte dynamique, des implantations parallèles sont réalisées afin d'accélérer les temps de réponse. Enfin, des tests numériques sont réalisés à l'aide de simulations qui s'inspirent de situations réelles observées dans les systèmes de courrier rapide. Les résultats numériques confirment la pertinence des différentes stratégies proposées.

Mots clés. répartition de véhicules, temps réel, diversion, temps de parcours variables dans le temps, anticipation, recherche avec tabous, parallélisme.

103-03106/

Table des matières

Sommaire i

Table des Matières iii

Liste des figures viii

Liste des tableaux x

Remerciements xii

Dédicace xv

1 Introduction 1

1.1 Motivation 1

1.2 Les problèmes de répartition dynamique de véhicules 3

1.3 Organisation de la thèse 6

2 Revue de littérature 12

2.1 Les premiers pas 14

2.2 Contextes d'application 15

2.2.1 Problèmes de camionnage 15

2.2.2 Problèmes de transport sur demande "dial-a-ride" 16

2.2.3 Problèmes de services de courrier rapide 17
2.2.4 Problèmes du réparateur 17
2.2.5 Problèmes de services d'urgence 18
2.2.6 Problèmes de routage et d'inventaire 18
2.3 Taxinomie 19
2.3.1 Degré de dynamisme 19
2.3.2 Présence de routes planifiées 22
2.3.3 Importance du repositionnement des véhicules 23
2.4 Méthodologies générales de résolution 24
2.4.1 Adaptation d'algorithmes statiques 25
2.4.2 Méthodes stochastiques 27
2.4.3 Méthodes basées sur la théorie des files d'attente 28
2.5 Les problèmes de cueillette (ou livraison) 28
2.5.1 Adaptation d'algorithmes statiques 29
2.5.2 Méthodes stochastiques 41
2.5.3 Les méthodes basées sur la théorie des files d'attente 43
2.6 Les problèmes mixtes de cueillette et livraison 45
2.6.1 Adaptation d'algorithmes statiques 45
2.6.2 Les approches stochastiques 56
2.6.3 Méthodes basées sur la théorie des files d'attente 58
2.6.4 Autres approches 59
2.7 Conclusion 60

Bibliographie **62**

3 Diversion Issues in Real-Time Vehicle Dispatching **74**

3.1 Literature review 78
3.1.1 Adaptation of static algorithms 78
3.1.2 Stochastic methods 80
3.1.3 Other methods 81
3.1.4 Diversion strategies 81
3.2 Problem definition 81
3.3 A parallel tabu search algorithm 83
3.4 A new strategy for handling customer requests 86
3.4.1 Motivation 86
3.4.2 A broader view 87
3.4.3 Solution quality versus computation time trade-off 89
3.5 Implementation 90
3.5.1 Inclusion of the current destination 90
3.5.2 Time projection 91
3.5.3 Handling new events 92
3.5.4 Setting δt 93
3.6 Computational results 94
3.6.1 Simulation framework 94
3.6.2 Numerical results 96
3.6.2.1 Preliminary tests 96
3.6.2.2 Parameter values 98
3.6.2.3 Assessing the new implementation 99
3.7 Conclusion 100

Bibliographie **101**

4 Vehicle Dispatching With Time-Dependent Travel Times 114

4.1 Introduction 116

4.2 Literature review 118

4.2.1 Models based on "simple" travel time and cost functions . . 120

4.2.2 Models based on discrete travel time and cost functions . . . 121

4.2.3 Models based on continuous travel time and cost functions . 121

4.2.4 Models based on Markovian formulations of travel time and cost functions 122

4.3 A time-dependent travel speed model 123

4.3.1 Motivation 123

4.3.2 The model 126

4.3.2.1 Changing travel speeds 127

4.3.2.2 Travel time calculation 128

4.3.2.3 Characteristics of the travel time function 128

4.4 A parallel tabu search algorithm 130

4.4.1 The original algorithm 130

4.4.2 Implementing the time-dependent model 132

4.4.2.1 Objective function 132

4.4.2.2 Neighborhood evaluation 132

4.5 Computational results 136

4.5.1 Test problems 136

4.5.2 Experiments 138

4.5.2.1 Preliminary tests 138

4.5.2.2 Numerical results 139

4.6 Dynamic problem 140

4.6.1 Problem definition 141
4.6.2 Original dynamic algorithm (Gendreau et al. (99)) 142
4.6.3 Time-dependent algorithm in a dynamic environment 142
4.6.4 Computational results 144
4.6.4.1 Simulation framework 144
4.6.4.2 Numerical results 145
4.7 Conclusion 146

Bibliographie **146**

5 Exploiting Knowledge about Future Demands for Real-Time Vehicle Dispatching **154**

5.1 Introduction 156
5.2 Literature review 158
5.2.1 Static stochastic vehicle routing problems 158
5.2.2 Dynamic stochastic vehicle routing problems 159
5.2.2.1 Analytical studies 159
5.2.2.2 Stochastic approaches 162
5.3 A probabilistic strategy for managing the fleet of vehicles 166
5.3.1 Motivation 166
5.3.2 The strategy 167
5.3.2.1 Basic assumptions 167
5.3.2.2 Waiting time 168
5.3.2.3 Probability of occurrence of a new request 169
5.3.2.4 Coverage of the territory 171
5.3.2.5 Summary of the strategy 171

5.4 A parallel tabu search algorithm 172
5.4.1 The original algorithm 172
5.4.2 Implementing the new strategy 174
5.5 Computational results 175
5.5.1 Simulation framework 175
5.5.1.1 Time horizon 176
5.5.1.2 Generation of new service requests 176
5.5.2 Testing scenarios 177
5.5.3 Experiments 179
5.5.3.1 Preliminary tests 179
5.5.3.2 Numerical results 180
5.6 Conclusion 189

Bibliographie **190**

6 Conclusion **194**

6.1 Principales contributions 194
6.2 Nouvelles avenues de recherche 196

Bibliographie **198**

Table des figures

1.1 La route planifiée d'un véhicule dans un environnement dynamique. 6

2.1 L'échange CROSS. 37

3.1 A vehicle route in a dynamic setting. 82

3.2 Diversion. 87

3.3 A dynamic assignment strategy that includes diversion. 88

3.4 Time projection 89

3.5 Update of solutions in the adaptive memory 91

4.1 Travel time function on a link. 124

4.2 Travel speed function at a node. 125

4.3 Changing travel speed over time. 127

4.4 Travel time calculation procedure. 128

4.5 An example of travel speed and travel time functions. 129

4.6 A CROSS exchange. 132

4.7 Approximate evaluation of lateness. 135

4.8 A vehicle route in a dynamic setting. 141

4.9 Finding the departure time for a vehicle in a dynamic environment. 143

4.10 Departure time calculation procedure in a dynamic environment. 144

5.1 Moving the vehicle far from an "interesting" area. 167

5.2 Defining the vehicle neighborhood $\bar{N}_i$. 170

5.3 Partitioning δt_i. 170

5.4 Pseudo-code of the proposed strategy. 171

5.5 A CROSS exchange. 173

Liste des tableaux

2.1 *Classification des problèmes de la section 2.2.* 25

3.1 Comparison of the refined implementation (new) with the old one using scenario 1. 102

3.2 Searching for the best parameter values under rule 2 for scenario 1 . 103

3.3 Searching for the best parameter values under rule 2 for scenario 2 . 104

3.4 Searching for the best parameter values under rule 3 for scenario 1 . 105

3.5 Searching for the best parameter values under rule 3 for scenario 2 . 106

3.6 Comparison with the original algorithm for scenario 1 107

3.7 Comparison with the original algorithm for scenario 2 108

4.1 *Travel speed matrix in scenario 1* 137

4.2 *Travel speed matrix in scenario 2* 137

4.3 *Travel speed matrix in scenario 3* 137

4.4 *Searching for the best value for parameter M* 147

4.5 *Comparison of time-dependent and constant speeds under the three scenarios in a static setting (with $M = 32$)* 148

4.6 *Comparison of time-dependent and constant speeds under the three scenarios in a dynamic setting (with $M = 32$)* 149

5.1 *Demand rate matrix in scenario 1.* 178

5.2 *Demand rate matrix in scenario 2.* 178

5.3 *Best parameter values with 4 vehicles for scenario 1* 183

5.4 *Best parameter values with 4 vehicles for scenario 2* 184

5.5 *Original algorithm under scenario 1 : Simulation of 5 hours with 4 vehicles and 27 requests per hour on average.* 185

5.6 *New algorithm under scenario 1 : Simulation of 5 hours with 4 vehicles and 27 requests per hour on average.* 185

5.7 *Original algorithm under scenario 1 : Simulation of 5 hours with 6 vehicles and 27 requests per hour on average.* 186

5.8 *New algorithm under scenario 1 : Simulation of 5 hours with 6 vehicles and 27 requests per hour on average.* 186

5.9 *Original algorithm under scenario 2 : Simulation of 5 hours with 4 vehicles and 36 requests per hour on average.* 187

5.10 *New algorithm under scenario 2 : Simulation of 5 hours with 4 vehicles and 36 requests per hour on average.* 187

5.11 *Original algorithm under scenario 2 : Simulation of 5 hours with 6 vehicles and 36 requests per hour on average.* 188

5.12 *New algorithm under scenario 2 : Simulation of 5 hours with 6 vehicles and 36 requests per hour on average.* 188

Remerciements

Je désire avant tout exprimer ma profonde gratitude à mes directeurs, les professeurs Michel Gendreau et Jean-Yves Potvin, pour leurs encouragements soutenus ainsi que pour le soutien financier qu'ils m'ont accordé tout au long de cette recherche. Je voudrais remercier le professeur Gendreau pour m'avoir suggéré un sujet gratifiant et m'avoir fait bénéficier de son expérience pertinente dans le domaine de tournées de véhicules. Qu'il trouve ici le témoignage de ma profonde estime. Je tiens aussi à remercier le professeur Potvin pour sa disponibilité et son support, ainsi que pour les remarques éclairées qu'il m'a prodiguées tout au long de cette recherche. J'espère sincèrement que cette thèse n'est que le début de collaborations futures tout aussi fructueuses.

Je profite de cette occasion pour remercier les professeurs, chercheurs, professionnels et étudiants du Centre de Recherche sur les Transports, pour avoir su créer une ambiance autant agréable qu'enrichissante. Je remercie tout particulièrement François Guertin pour sa disponibilité et son aide lors de mes premiers pas en programmation parallèle. Je remercie également l'équipe du secrétariat du centre, en particulier mesdames Lucie-Nathalie Cournoyer, Lucie l'Heureux et Martine Gemme, pour leur patience et les nombreux services rendus, toujours avec le sourire, tout au long de ces années d'études.

Je rends hommage à tous mes ami(e)s qui par leur présence chaleureuse et par leur soutien moral, ont su rendre ces longues dernières années agréables malgré les moments difficiles.

Finalement, tout mon respect et mon dévouement vont à ma famille qui m'a

toujours soutenu malgré les grandes distances. Je remercie en particulier mes parents pour leurs encouragements soutenus. Je remercie également mon frère aîné, ami et confident Abderrahmane qui a toujours été si près de moi tout en étant de l'autre côté de l'atlantique. Je réserve mon mot de la fin à une pensée toute particulière à mon défunt père.

À la mémoire de mon père Mohammed Ben taleb Ichoua

À ma mère Touria El Alami

Chapitre 1

Introduction

1.1 Motivation

L'économie mondiale moderne s'oriente de plus en plus vers l'ouverture des marchés locaux et encourage les échanges de ressources sur l'ensemble du globe. Cette nouvelle tendance entraîne une plus grande compétitivité dans la recherche de l'efficacité à tous les niveaux, entre autres dans la distribution des biens et services. Ainsi, des problèmes de gestion de flottes de véhicules très complexes et de taille considérable sont apparus. En conséquence, toute amélioration dans les techniques de résolution de ces problèmes a un impact majeur sur le plan économique. On peut diviser ces problèmes en deux grandes classes : les problèmes statiques et les problèmes dynamiques. Dans les problèmes statiques, l'information est connue à l'avance et reste invariante dans le temps. Par contre, dans les problèmes dynamiques, l'information est incomplète et est sujette à des variations aléatoires dans le temps. Dans ce dernier cas, le planificateur a la tâche ardue de prendre des décisions dans un environnement dynamique où les événements se produisent de façon aléatoire dans le temps. Cette charge est d'autant plus

pénible que les décisions doivent se prendre dans un temps limité, en se basant sur des informations incertaines et qui évoluent dans le temps. Toutefois, de nos jours, les technologies de l'information et de la communication (p.e. positionnement par satellite, système d'information géographique, échanges informatisés des données, télé-détection,...) permettent d'obtenir une grande quantité d'informations en temps réel. Ces dernières peuvent alors être exploitées par la procédure de résolution. D'autre part, les progrès récemment accomplis dans le domaine de l'informatique, et en particulier en calcul parallèle, offrent des opportunités inestimables pour l'élaboration de méthodologies rapides capables de transiger avec la composante "temps réel" du problème. Tous ces développements technologiques ont certes permis au domaine de la répartition dynamique de véhicules de prendre son élan. Toutefois, ce champ est encore loin d'avoir atteint sa phase de maturité. En effet, relativement peu de travaux ont été réalisés dans ce domaine. Pourtant, plusieurs problèmes concrets qui constituent le moteur de l'économie moderne s'y rattachent. Nous citons entre autres, les services de poste prioritaire [4], la livraison de produits pétroliers et de gaz industriels [2], la gestion des conteneurs, ainsi que les problèmes de transport sur demande ("dial-a-ride") comme le transport adapté et le transport scolaire [7].

Ces besoins économiques et stratégiques, appuyés par les nouveaux moyens technologiques évoqués plus haut, nourrissent la recherche visant le développement de systèmes d'aide à la décision qui sont en mesure de fournir de bonnes solutions, conformes à la réalité. Notre thèse s'inscrit dans le cadre de cette nouvelle direction de recherche et puise sa motivation plus particulièrement dans la volonté de répondre à quelques-unes des problématiques rencontrées dans des applications temps-réel. Ce but est atteint en deux étapes. La première étape consiste à

présenter ces problématiques et à les analyser. La seconde étape vise à élaborer des stratégies rapides et efficaces afin de les résoudre.

Dans ce qui suit, nous décrivons le problème de répartition de véhicules en temps réel et nous présentons l'organisation générale de la thèse.

1.2 Les problèmes de répartition dynamique de véhicules

Limitées par les moyens techniques et méthodologiques, les recherches antérieures se sont majoritairement consacrées à l'étude de l'approximation statique de ces problèmes [1, 3]. Sous leur forme générale, ces problèmes se décrivent comme suit. Considérons un ensemble de clients dont chacun requiert un service sur place (problème avec requêtes à un point), ou un service de transport d'un point de collecte vers un point de livraison (problème avec requêtes à deux points). Chacun des points de service possède une fenêtre de temps et un temps de service, et pour les servir, nous disposons d'une flotte de véhicules de capacité finie. Il s'agit alors d'identifier un ensemble de routes de coût minimal qui soient réalisables (c.à.d., qui satisfont les contraintes de capacité des véhicules et les fenêtres de temps). Par ailleurs, les fenêtres de temps des points de service peuvent être rigides ou souples. Dans le premier cas, on doit respecter les bornes inférieure et supérieure de la fenêtre à chaque point de service. Dans le deuxième cas, on permet la violation de la borne supérieure moyennant une pénalité dans la fonction objectif, et selon les différentes versions, on permet ou non la violation de la borne inférieure moyennant un temps d'attente. Dans le cas des problèmes avec requêtes à un point, les origines ou les destinations de toutes les requêtes se confondent et correspondent au dépôt.

Notons enfin que les problèmes sans fenêtres de temps découlent de la définition précédente en associant à chaque point la fenêtre $]-\infty, +\infty[$.

Dans le cas statique, les informations comme les demandes des clients ou les temps de parcours et de service sont connus avant la construction des routes, et sont invariants. En revanche, dans le cas dynamique, l'information disponible au départ est incomplète et est sujette à des variations aléatoires dans le temps (p.e. nous ne connaissons qu'un nombre limité de requêtes, les temps de parcours varient à cause de la congestion sur le réseau, etc.). Ainsi, l'information est dévoilée au fur et à mesure que nous construisons les routes, ajoutant ainsi une difficulté additionnelle au problème.

Notons enfin que si tout problème dynamique possède un caractère stochastique, en revanche, un problème stochastique n'est pas forcément dynamique. Par exemple, dans le problème du voyageur de commerce probabiliste [5, 6], bien qu'il y ait de l'incertitude concernant la présence ou non d'un client sur une route, la probabilité liée à la présence de ce client est connue à l'avance. Ainsi, l'information nécessaire à l'élaboration d'une politique optimale (en terme d'espérance de la distance totale parcourue) est disponible dès le départ. Nous concluons alors que le problème est statique.

Notons que les sujets que nous abordons dans notre recherche sont applicables aussi bien aux problèmes avec requêtes à un point qu'à celles à deux points. Toutefois, pour fins de simplicité, nous avons considéré une application qui correspond à un problème avec requêtes à un point, soit la partie locale d'un service de courrier international, comme Federal Express (où l'on recueille le courrier chez différents clients pour le retourner ensuite à un centre de traitement central). La version statique de ce problème se décrit comme suit. Considérons une flotte homogène

de véhicules et un ensemble de requêtes de service. Le but est de déterminer un ensemble de routes de coût minimal qui desservent ces requêtes tout en satisfaisant les contraintes suivantes :

- Chaque véhicule commence sa tâche à partir d'un dépôt auquel il revient en fin de tournée.
- Chaque véhicule est affecté à une seule route et le point de service de chaque requête est visité une et une seule fois, par un et un seul véhicule.
- Il n y a pas de contraintes de capacité (puisque les objets transportés, lettres ou colis, sont de petite taille).
- Chaque route doit respecter la fenêtre de temps au dépôt (qui correspond à la période de disponibilité des véhicules, qui est la même pour tous). En revanche, les fenêtres de temps associées aux requêtes sont souples. Ainsi, un véhicule peut arriver au point de service avant la borne inférieure ou après la borne supérieure de la fenêtre de temps. Dans le premier cas, il est tenu d'attendre. Dans le second cas, le retard résultant est pénalisé dans la fonction objectif.

L'objectif est de minimiser une somme pondérée de la distance totale parcourue par les véhicules et du retard total sur l'ensemble des requêtes.

Dans le cas dynamique, certaines requêtes ne sont pas connues à l'avance mais sont plutôt révélées en temps réel. Dans un environnement dynamique, à chaque instant t, la route d'un véhicule donné peut être divisée en trois parties (voir la figure 1.1) :

- les déplacements complétés, qui forment la partie de la route que le véhicule a déjà parcourue. De ce fait, cette partie ne peut être modifiée ;
- le déplacement courant du véhicule ;

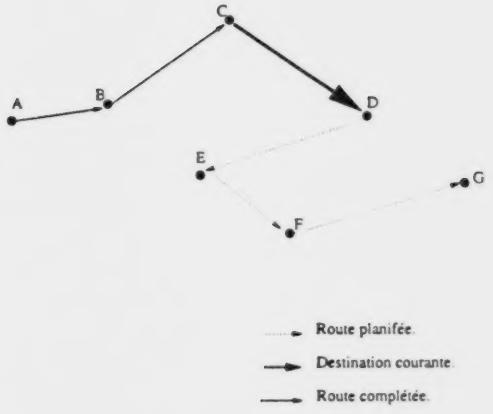

FIG. 1.1 – La route planifiée d'un véhicule dans un environnement dynamique.

- les déplacements planifiés, qui constituent la partie de la route non encore exécutée par le véhicule (route planifiée).

Lorsqu'une nouvelle requête apparaît, il s'agit de l'insérer à coût minimal, dans la route planifiée d'un véhicule donné.

1.3 Organisation de la thèse

Outre le présent chapitre, le corps de la thèse se compose de cinq chapitres, chacun pouvant être lu indépendamment des autres. De ce fait, la bibliographie relative à chacun d'eux, est indiquée à la fin du chapitre. Le chapitre 2 présente une revue très large de la littérature portant sur les problèmes de répartition et routage dynamique de véhicules. Dans cette revue, nous nous intéressons tout particulièrement à l'aspect "temps réel" et nous proposons une taxinomie permettant de classifier ces problèmes sous cet aspect.

Les trois chapitres suivants, qui correspondent à des articles publiés ou soumis à des revues scientifiques, se concentrent de façon plus spécifique sur des

problèmes avec requêtes à un point rencontrés dans les services de courrier rapide (où l'on recueille le courrier chez différents clients pour le retourner ensuite à un centre de traitement). En fait, chacun de ces chapitres se penche sur des problématiques particulières touchant à l'aspect temps réel de ces problèmes, problématiques qui ont été peu ou pas abordées dans la littérature. Celles-ci se rapportent soit à une modélisation plus réaliste des problèmes rencontrés en pratique, soit à l'intégration et la reproduction de pratiques de répartition souvent observées chez les répartiteurs humains, et qui sont désirables pour améliorer la qualité des solutions produites. Dans chacun des cas, une plate-forme commune a été utilisée afin de valider nos hypothèses. Il s'agit en l'occurrence d'un algorithme de recherche tabou, développé antérieurement par Gendreau et al.[4], et que nous avons modifié et adapté pour les besoins de chacune des études.

Dans le chapitre 3, nous nous intéressons à la diversion, soit la possibilité de modifier la destination courante d'un véhicule afin de desservir une nouvelle requête qui apparaît à proximité de ce véhicule. Notons qu'il s'agit d'une pratique courante chez les répartiteurs. Afin de réaliser et étudier les bénéfices de la diversion, nous avons dû modifier l'algorithme tabou de base en étendant le voisinage de façon à y inclure la destination courante du véhicule (destination qui était fixée et inamovible dans la version originale de l'algorithme).

Dans le chapitre 4, nous abordons un modèle où l'on permet des temps de parcours qui varient selon la période de la journée. Ceci permet évidemment de mieux s'approcher des applications réelles où les temps de parcours sont tout naturellement sujets à des variations dans le temps. La principale particularité de notre modèle est l'utilisation explicite de la vitesse des véhicules. Cette vitesse est modifiée lors du parcours d'un lien par un véhicule, si ce dernier part de l'origine à une

certaine période pour arriver à destination à une période différente. Ceci constitue un raffinement important par rapport aux modèles proposés antérieurement, où le temps de parcours et la vitesse des véhicules sont constants sur tout le lien, et déterminés exclusivement en fonction de la période où le véhicule quitte l'origine. Grâce à ce raffinement, il est maintenant assuré qu'un véhicule partant plus tôt du sommet origine arrivera plus tôt à destination (ce qui constitue une caractéristique hautement désirable). Le modèle a été évalué en modifiant la version originale de l'algorithme de recherche tabou, qui ne considérait que des temps de parcours invariants.

Dans le chapitre 5, nous nous intéressons à une question extrêmement importante. Comment exploiter toute connaissance probabiliste que nous possédons sur les événements futurs, en particulier les requêtes à venir, afin de produire des solutions plus robustes et de meilleure qualité au temps courant ? L'algorithme de recherche tabou original était très limité à cet égard, en ce sens qu'il se contentait d'optimiser sur la base des requêtes déjà reçues (et donc, connues avec certitude). Nous avons réduit considérablement la myopie de cet algorithme, en introduisant des demandes fictives qui correspondent à des requêtes prévues, mais non encore réalisées. Grâce à cette technique, nous pouvons maintenant produire des routes planifiées pour les véhicules qui sont plus robustes, car leur orientation et leur forme générale tiennent compte des requêtes à venir. Évidemment, des mécanismes appropriés de génération et d'élimination de requêtes fictives (en cas de non réalisation) ont aussi été développés dans le cadre de ce travail.

Finalement, dans le chapitre 6 nous soulignons les principales contributions de cette thèse et nous proposons quelques avenues de recherche qu'il serait pertinent d'explorer.

Il importe de noter que les tests numériques rapportés dans cette thèse sont réalisés à l'aide de simulations qui se rapprochent autant que possible de situations réelles observées dans les systèmes de courrier rapide. Dans ces applications, on retrouve en général un certain nombre de requêtes statiques (qui correspondent à des requêtes reçues à la fin de la journée précédente). Dans nos simulations, 25 à 50 % des requêtes sont statiques. Par ailleurs, le nombre total des clients varie entre 100 et 155 au cours d'une journée et la taille de la flotte est de 4 à 10 véhicules. L'intensité d'arrivée des requêtes est de 3 à 5 requêtes à la minute. Il faut toutefois noter que dans les applications réelles, on observe une grande variance dans ces caractéristiques selon l'importance de la compagnie considérée, la taille de sa flotte, le territoire qu'elle couvre, etc...

Bibliographie

[1] **M.O.Ball, T.L. Magnanti, C.L. Monna and G.L. Nemhauser (eds),** *Network Routing*, Handbooks in Operations Research and Management Science, Volume 8, North-Holland : Amsterdam (1995).

[2] **W.J. Bell, L.M. Dalberto, M.L. Fisher, A.J. Greenfield, R. Jaikumar, P. Kedia, R.G. Macj, P.J. Prutzman,** *Improving the Distribution of Industrial Gases with an On-Line Computrized Routing and Scheduling Optimizer* , Interfaces **13** (1983), 04-23.

[3] **J. Desrosiers, Y. Dumas, M.M. Solomon and F. Soumis,** *Time Constrained Routing and Scheduling*, in Handbooks in Operations Research and Management Science, Volume 8 : Network Routing, M.O. Ball, T.L. Magnanti, C.L. Monma and G.L. Nemhauser (eds), pp. 35-139, North-Holland : Amsterdam (1995).

[4] **M. Gendreau, F. Guertin, J.Y. Potvin and É. Taillard,** *Parallel Tabu Search for Real-Time Vehicle Routing and Dispatching*, Transportation Science **33 (4)** (1999), 381-390.

[5] **P. Jaillet,** *The Probabilistic Traveling Salesman Problem*, Ph.D. Thesis, Department of Civil Engineering, MIT (1985).

[6] **P. Jaillet,** *Probabilistic Routing Problems in the Plane*, Operational Research **29** (1991), 675-688.

[7] **H.N. Psaraftis,** *A Dynamic Programming Solution to the Single-Vehicule Many-to-Many Immediate Request Dial-A-Riade Problem* , Transportation Science **14** (1980), 130-154.

Chapitre 2

Revue de littérature

En dépit de sa richesse et de l'abondance des travaux qui lui sont consacrés, le problème de répartition de véhicules en temps réel ne représente qu'un sous-ensemble d'une plus large famille communément appelée "Problèmes dynamiques de transport". Cette vaste classe englobe entre autres les problèmes dynamiques de plus courts chemins, d'affectation de trafic, de contrôle aérien ainsi que le problème dynamique de localisation (voir Dror et Powell [21]). Dans ce chapitre, nous passons en revue les travaux portant sur le problème de répartition de véhicules en temps réel. En premier lieu, nous présentons un bref historique du problème. En second lieu, nous dressons un inventaire non exhaustif des problèmes qui s'apparentent à ce domaine. Dans la troisième partie, nous proposons une taxinomie basée sur les attributs de ces différents problèmes. Par la suite, nous utilisons cette taxinomie afin d'identifier certaines caractéristiques communes à des problèmes qui peuvent sembler très différents à première vue. Dans la quatrième partie, nous abordons les méthodologies générales adoptées pour traiter l'aspect dynamique de ces problèmes et nous enchaînons avec une discussion de ces approches. Dans les cinquième et sixième parties, nous présentons les travaux portant sur les problèmes

de cueillette (ou livraison) et les problèmes mixtes avec cueillette et livraison, respectivement. Une classification basée sur les approches de résolution est adoptée pour chaque type de problèmes, soit l'adaptation d'algorithmes statiques, les techniques stochastiques ou les méthodes basées sur la théorie des files d'attente. Nous concluons chaque partie par une brève discussion des avantages et des faiblesses des approches proposées. Dans la dernière partie, nous présentons nos conclusions et nos remarques générales relatives aux travaux présentés dans ce chapitre.

2.1 Les premiers pas

Les premiers travaux portant sur la répartition de véhicules en temps réel ont été accomplis vers la fin des années 70 [17, 85, 93, 94]. L'objectif principal de ces travaux était de prédire la performance des systèmes de transport sans avoir recours aux méthodes de simulation encore coûteuses à cette époque. En conséquence, les chercheurs se sont consacrés à l'élaboration de modèles mathématiques simples auxquels ils ont appliqué des règles très intuitives (p.e. après chaque arrêt, se diriger vers la destination la plus proche parmi les destinations des passagers, etc...). Ces approches simplistes visaient d'une part à mesurer la performance des systèmes de transport, et d'autre part à amorcer des études analytiques et paramétriques permettant de cerner les caractéristiques principales de ces systèmes. A titre d'exemples, nous citons le travail de Daganzo [17] et celui de Stein [85] qui ont été réalisés en 1978. Les deux chercheurs se sont penchés sur un problème de transport de personnes sur demande ("dial-a-ride") où les requêtes arrivent en temps réel. Ce problème se range dans la catégorie des problèmes où chaque requête est constituée d'un point de cueillette et d'un point de livraison (requête à deux points). Ces deux travaux se basent sur des modèles de files d'attente et de processus markoviens. Dans le premier travail, l'auteur fournit une estimation du temps moyen d'attente et du temps passé dans le système, en fonction du nombre estimé des clients. Dans le second travail, et dans le cas d'un seul véhicule, l'auteur prouve l'optimalité asymptotique d'une procédure simple de construction qui minimise le temps total écoulé entre l'apparition d'une requête et sa satisfaction. L'auteur généralise ensuite son résultat au cas où plusieurs véhicules sont disponibles.

La faiblesse de ce genre d'études réside dans leur saveur hautement théorique

entraînant une difficulté d'application dans des cas concrets. D'une part, l'optimalité asymptotique suppose que le nombre de clients est infiniment grand, ce qui pose le problème de détermination d'un seuil à partir duquel les résultats peuvent s'appliquer. D'autre part, dans le cas multi-véhicules, on suppose qu'on dispose d'une division de la région étudiée en sous-régions, ainsi qu'une affectation préalable des véhicules à ces différentes sous-régions. Or ces deux composantes sont au coeur de la complexité du problème. Retenons toutefois que ces études théoriques ont permis une nouvelle compréhension et un intérêt nouveau pour le problème. Vers la fin des années 80, les progrès dans les technologies de communications ont donné un nouveau souffle au domaine de la répartition et du routage de véhicules en temps réel. Ainsi, une nouvelle génération de méthodes permettant d'exploiter ces technologies a vu le jour [45, 55, 69].

2.2 Contextes d'application

La majorité des applications reliées au problème de répartition de véhicules en temps réel, peuvent être classées de la façon suivante :

2.2.1 Problèmes de camionnage

Dans ce contexte, il s'agit d'affecter une flotte de camions en mouvement à des requêtes qui apparaissent en temps réel dans une zone géographique donnée. Étant donné que cette zone est souvent étendue (p.e. contexte régional ou national), le temps de service d'une requête donnée est relativement long (quelques jours). En outre, chaque requête est caractérisée par un point de collecte, un point de livraison, et éventuellement une fenêtre de temps associée à chacun de ces deux points. De manière très synthétique, on peut distinguer deux types de problèmes

de répartition dynamique de camions : les problèmes avec consolidation, ou selon la terminologie anglaise "less-than-truckload problems" (LTL) [3, 4, 16], et les problèmes sans consolidation, ou selon la terminologie anglaise "truckload trucking problems" (TL) [24, 48, 54, 56, 58, 59, 60, 72, 73, 74, 75, 91]. Dans les problèmes du premier type, les demandes de plusieurs clients différents peuvent être transportées par un même véhicule. Dès lors, la construction de routes optimales permettant de servir ces différents clients a un impact majeur sur l'efficacité de l'approche de résolution. En revanche, dans les problèmes du second type, un camion ne transporte qu'une seule cargaison à la fois. De ce fait, ces problèmes sont plus faciles que ceux du premier type et se résument à l'affectation des camions libres aux requêtes non encore satisfaites.

2.2.2 Problèmes de transport sur demande "dial-a-ride"

Dans ces problèmes, une flotte de véhicules avec contraintes de capacité assurent le transport de clients ayant des besoins particuliers (p.e. personnes âgées ou handicapées, transport scolaire, etc...). Chaque client a un point de collecte, un point de livraison, et éventuellement une fenêtre de temps associée à ces deux points. Par ailleurs, les contraintes de capacité des véhicules doivent être respectées.

Notons qu'au cours des trente dernières années, les besoins accrus dans le domaine du transport adapté (c-à-d., transport de personnes âgées ou handicapées) ont mené à plusieurs travaux dans ce domaine [17, 20, 46, 61, 62, 63, 66, 76, 79, 85, 93, 94].

2.2.3 Problèmes de services de courrier rapide

Les applications s'apparentant à ces problèmes s'inscrivent dans deux contextes différents : services de courrier rapide international [28] et services de courrier rapide urbain [78, 27, 44, 82, 6]. Dans le premier contexte, on s'intéresse exclusivement à la phase locale du processus de traitement du courrier rapide international (p.e. FedEx, UPS) où une flotte de véhicules en mouvement est utilisée afin de recueillir le courrier chez différents clients et le ramener ensuite à un centre de traitement. Notons que des fenêtres de temps souples ou rigides peuvent être associées aux points de collecte ainsi qu'au centre de traitement. Dans le second contexte, chaque requête de service a un point de collecte et un point de livraison situés dans un même secteur urbain. De ce fait, outre les contraintes de fenêtres de temps, les points de collecte et de livraison de chaque requête doivent être visités par le même véhicule avec une contrainte de précédence entre les deux points.

2.2.4 Problèmes du réparateur

Le problème du réparateur est l'un des problèmes de répartition de véhicules en temps réel les plus étudiés [7, 8, 9, 87]. Dans ce problème, une flotte de véhicules doit visiter des clients aléatoirement répartis dans une zone géographique donnée afin de satisfaire leurs requêtes pour un service de réparation. Une fois chez son client, chaque véhicule passe un temps de service d'une durée stochastique (relativement longue) avant d'être en mesure de satisfaire la prochaine requête parmi celles qui sont encore pendantes. Plusieurs applications réelles s'inscrivent dans ce cadre. Notons entre autres, les compagnies de services publics (p.e., les compagnies d'électricité).

2.2.5 Problèmes de services d'urgence

Dans ces problèmes, une flotte de véhicules doit satisfaire la demande d'un ensemble de clients provenant d'une zone géographique donnée (p.e. ambulances, taxis, pompiers). Ces problèmes sont caractérisés par l'urgence des requêtes et le fait que chaque véhicule est dédié à un client particulier [30, 90].

2.2.6 Problèmes de routage et d'inventaire

Dans ces problèmes, une flotte de véhicules assure l'approvisionnement d'un ensemble de clients dont la demande n'est pas connue à l'avance avec précision [51]. Chaque véhicule possède une capacité finie et doit revenir à un dépôt central en fin de journée. Par ailleurs, chaque client spécifie une fenêtre de temps durant laquelle il préfère être servi. Plusieurs applications concrètes s'apparentent à cette classe de problèmes tels les problèmes de livraison de produits pétroliers [3, 4, 16]. Notons que dans ces problèmes, la quasi-totalité des clients sont connus à l'avance. D'autre part, la prévision des demandes futures des clients est relativement précise ce qui réduit la probabilité d'occurrence d'événements inattendus auxquels il faut réagir très rapidement (p.e. incapacité de satisfaire la totalité de la demande du client courant).

Nous avons tenté ici de dresser un inventaire non exhaustif des différentes applications s'inscrivant dans le contexte de la répartition dynamique d'une flotte de véhicules. Cet exercice nous a révélé l'étendue de ce domaine où une multitude d'applications, prenant naissance dans des contextes très variés, tantôt partagent des traits en commun et tantôt diffèrent par certains aspects. Il serait donc utile d'établir une taxinomie qui permettrait de mieux comprendre, voire mieux définir, ces différents problèmes. Un pas important serait alors franchi vers

le développement de méthodologies appropriées pour chaque classe de problèmes. Dans la prochaine section, nous explorons ce sujet.

2.3 Taxinomie

L'intérêt majeur d'une taxinomie est qu'elle permet d'identifier une structure commune à des problèmes qui sont en apparence très différents [57]. Cette classification n'est certes qu'une étape préliminaire, mais combien cruciale à l'élaboration de méthodologies et d'approches de résolution appropriées pour chaque groupe de problèmes partageant une structure commune. Dans ce qui suit, nous proposons de classifier les problèmes décrits dans la section 2.2 selon les caractéristiques suivantes :

1- Degré de dynamisme.

2- Présence de routes planifiées.

3- Importance du repositionnement des véhicules.

Le reste de la section sera consacré à une discussion détaillée de cette taxinomie.

2.3.1 Degré de dynamisme

Le "degré de dynamisme" d'un problème dépend de deux dimensions importantes. La première dimension est la fréquence des changements affectant les données au cours du temps alors que la deuxième est la rapidité avec laquelle il faut prendre des décisions suite à l'occurrence de ces changements.

(i) Fréquence des changements affectant les données

Les changements affectant les données au cours du temps sont de deux types. Le premier type concerne la fréquence d'apparition de nouvelles requêtes. Ainsi, les problèmes où les nouvelles requêtes sont peu fréquentes ont un faible degré de

dynamisme et sont de ce fait très semblables à leur version statique.

Le second type concerne le degré de "stochasticité" des attributs d'une requête nouvellement apparue (p.e. date de début de service, date d'arrivée, quantité de la demande, etc...). Ainsi, plus ces attributs varient dans le temps, plus le degré de dynamisme du problème considéré est élevé. À l'opposé, plus ces attributs demeurent stables dans le temps, plus l'écart entre la version dynamique du problème et sa version statique se réduit.

(ii) Degré d'urgence de prendre une décision

L'urgence de prendre une décision concernant une requête nouvellement apparue est liée au délai entre la date d'apparition de cette requête et le temps de début de service. Ainsi, une requête est d'autant plus urgente que le délai est plus court. Soulignons qu'en présence de fenêtres de temps, ce délai est d'autant plus court que la fenêtre de temps de la requête est proche du temps courant.

Le degré d'urgence des requêtes apparaissant en temps réel joue un rôle déterminant dans le degré de dynamisme d'un problème. En effet, même si la fréquence d'apparition des requêtes est très élevée, si ces requêtes ne sont pas urgentes, l'exécution des routes peut se faire bien après leur construction. Ainsi, des délais relativement longs peuvent être exploités afin de retarder l'exécution des routes le plus possible en vue de s'ajuster à d'éventuels changements. Le problème devient alors très semblable à la version statique où la simultanéité de la construction des routes et de leur exécution n'est pas requise.

Dans ce qui suit, nous nous basons sur le critère du degré de dynamisme afin de classer (de manière très synthétique) les différents problèmes présentés à la section 2.2. Il importe de noter que ce classement n'est produit qu'à titre indicatif et ne peut englober toutes les complexités et les particularités propres à chaque

application réelle.

1- Les problèmes ayant le degré de dynamisme le plus faible sont les problèmes de routage et d'inventaire, les problèmes de services de réparation à domicile ainsi que les problèmes de transport adapté. En effet, dans les problèmes de routage et d'inventaire, la quasi-totalité des clients sont connus à l'avance et la probabilité qu'un changement inattendu survienne dans les données est très faible (voir la sous-section 2.2.5). De ce fait, ces problèmes sont proches de la famille des problèmes statiques.

Dans les problèmes de services de réparation à domicile ainsi que dans les problèmes de transport adapté, les requêtes de service auxquelles il faut répondre en temps réel sont assez peu fréquentes. Par ailleurs, dans les services de réparation à domicile les fenêtres de temps sont souvent définies par la compagnie plutôt que par les clients. Il en découle que l'exécution des routes se fait bien après leur construction. Ainsi, l'impact de la composante "temps réel" du problème est très réduit et se résume en grande partie à la nécessité d'informer le client assez rapidement de la tranche de temps où se situera sa visite [47].

Dans le cas du transport adapté, les personnes faisant appel à ces services doivent planifier leurs déplacements avant les dates prévues pour ces derniers. Ainsi, le pourcentage de requêtes apparaissant en temps réel est généralement très réduit.

2- Les problèmes ayant un degré de dynamisme moyen sont les problèmes de camionnage. Dans ces problèmes, près de 60% du nombre total de requêtes sont révélées en temps réel [53] avec des délais qui peuvent être d'une durée de un à quelques jours. Toutefois, le degré d'urgence de la prise de décision dépend de la politique adoptée par la compagnie. En effet, dans certains cas, afin d'affronter la concurrence, le planificateur est tenu d'informer le client assez rapidement de

la décision d'acceptation ou de rejet de sa requête. Parfois, cette décision doit se prendre en l'espace de quelques minutes.

3- Les applications dont le degré de dynamisme est le plus élevé incluent les services de courrier rapide, les services de taxis et les services d'urgence comme les systèmes de gestion des ambulances, les patrouilles de police et les services de pompiers. Ces applications prennent place dans des zones géographiques de dimension limitée (p.e. contexte urbain). Dès lors, les contraintes temporelles sont plus strictes que dans les applications de camionnage par exemple, limitant ainsi les délais associés aux requêtes (quelques secondes à quelques minutes). D'autre part, les attributs temporels de ces problèmes (p.e., temps de parcours, fenêtres de temps et temps de service) sont sujets à de fréquentes variations stochastiques. Dans ces applications, il arrive aussi que le système devienne rapidement saturé vu que la quasi-totalité des requêtes apparaissent en temps réel et sont rapprochées dans le temps. Par exemple, le système de véhicules d'urgence de Montréal affecte un véhicule à toutes les 2.3 minutes et le délai entre la réception de l'appel et l'arrivée chez le client est généralement de l'ordre de 5 à 10 minutes [15].

2.3.2 Présence de routes planifiées

La présence de routes planifiées est un autre critère permettant ce classifier les différents problèmes de répartition de véhicules en temps réel. Nous définissons la route planifiée d'un véhicule comme étant une séquence ordonnée de requêtes reçues et affectées à ce véhicule, mais qui n'ont pas encore été desservies. L'ordre de la séquence est basé sur les dates prévues pour le début de service à ces requêtes. Des routes planifiées apparaissent lorsque la consolidation de plusieurs requêtes sur un même véhicule est permise.

Dans un contexte dynamique où les données sont sujettes à des variations souvent aléatoires dans le temps, l'effort de construire des routes planifiées pour les véhicules est discutable. Cet effort peut être légitime si la longueur de l'horizon de temps considéré est relativement courte (p.e. services de courrier urbain) ou si les requêtes ne sont pas très urgentes (p.e. services de réparation à domicile). Dans le cas contraire, un nombre non négligeable d'événements est susceptible de se produire entre le moment où une requête est insérée dans la route planifiée d'un véhicule et le moment où cette requête est effectivement desservie par le véhicule, rendant ainsi le plan rapidement désuet.

2.3.3 Importance du repositionnement des véhicules

Cette question joue un rôle central dans la performance des approches de résolution pour des problèmes de répartition dynamique de véhicules sans routes planifiées. En fait, les routes planifiées offrent une orientation générale des itinéraires des véhicules. Ainsi, lorsqu'un véhicule finit sa tâche courante, sa route planifiée suggère une prochaine destination possible. Cependant, dans certains problèmes où les durées des tâches sont très longues, la construction de routes planifiées est inutile (voir la sous-section 2.3.2). Par ailleurs, d'autres applications telles que les services d'urgence, n'autorisent pas la consolidation des requêtes au sein d'un même véhicule. Dans ces applications, la question de repositionner adéquatement les véhicules dans l'attente d'une prochaine demande une fois leur tâche courante complétée, joue un rôle crucial dans l'efficacité de la méthodologie de résolution. Il importe de noter que les critères de la "présence de routes planifiées" et de "l'importance du repositionnement des véhicules" sont duals l'un par rapport à l'autre. En effet, dans les problèmes où l'on construit des routes planifiées, on ne

s'intéresse généralement pas au repositionnement des véhicules et vice versa. De ce fait, dans ce qui suit nous représentons ces deux critères par une seule "fonction", notée R/P, ayant deux valeurs possibles, soient la valeur P correspondant au critère de "présence de routes planifiées" et la valeur R correspondant au critère de "l'importance du repositionnement des véhicules"

Dans ce qui suit, nous nous basons sur la taxinomie proposée afin de classifier (de manière très synthétique) les différents problèmes présentés à la section 2.2. Il importe de noter que cette classification ne peut en aucun cas englober toutes les complexités et les particularités propres à chaque application réelle. Le tableau 2.1 illustre la classification proposée. Dans ce tableau, les lignes correspondent aux trois niveaux de degré de dynamisme considérés, soient faible, moyen et élevé. Par ailleurs, les colonnes correspondent aux deux valeurs possibles de la fonction R/P. Notons que la case située à l'intersection de la première ligne et la troisième colonne du tableau ne contient aucune des applications décrites à la section 2.2. Ceci peut être attribué à la non pertinence du repositionnement des véhicules libres quand le degré de dynamisme est faible.

2.4 Méthodologies générales de résolution

Dans la littérature, nous distinguons trois approches principales pour le traitement de l'aspect dynamique des problèmes. La première est l'adaptation d'algorithmes initialement élaborés pour des problèmes statiques. La seconde repose sur des techniques de modélisation et de programmation stochastiques. La troisième est basée sur la théorie des files d'attente. Dans ce qui suit, nous décrivons chacune de ces trois catégories de méthodes. Nous soulignons également les points forts et les points faibles de chacune.

Dynamisme \ R/P	P	R
Faible	. Routage et inventaire . Transport adapté . Transport scolaire . Service de réparation à domicile	
Moyen	. Camionnage avec consolidation (LTL)	. Camionnage sans consolidation (TL)
Élevé	. Services de courrier rapide	. Services de taxis . Services publics d'urgence . Services de réparation d'urgence

TAB. 2.1 – *Classification des problèmes de la section 2.2.*

2.4.1 Adaptation d'algorithmes statiques

Cette approche repose sur la notion d'horizon de temps fuyant ("rolling horizon") [67]. Il s'agit ici de résoudre un problème statique défini sur les données connues avec certitude à l'intérieur de l'horizon fuyant, à chaque fois qu'un nouvel événement survient et modifie les données. La difficulté majeure de cette approche est le temps de calcul exigé par une exécution répétitive d'un algorithme statique à des intervalles de temps très courts (p.e., changement très fréquent des données). Afin de contourner la pression temporelle inhérente à un contexte dynamique, il est possible d'appliquer un algorithme statique complexe en phase d'initialisation sur les requêtes connues à priori, et d'avoir recours à des opérations rapides d'ajustement local lorsqu'un changement survient. Il importe de noter que l'impact de l'algorithme statique en phase d'initialisation diminue au fur et à mesure que le degré de dynamisme du problème augmente. Lorsque le degré de dynamisme est

très élevé, la qualité de la solution obtenue dépend principalement de l'efficacité des opérations de mises à jour locales utilisées. Les plus populaires sont les techniques d'insertion, jumelées parfois à des techniques d'amélioration de la solution. Parmi ces dernières, nous soulignons les méthodes d'échanges de requêtes comme celles proposées par Psaraftis [62] dans un contexte de transport sur demande. Toutefois, l'application de ces méthodes d'échange nécessite parfois des temps de calcul élevés, en particulier lors de l'évaluation des nouvelles solutions obtenues. D'autres chercheurs profitent du temps disponible entre deux événements pour appliquer des approches encore plus sophistiquées de type métaheuristique. Gendreau et al. [28] ont adopté cette approche dans un contexte de courrier rapide. Tout comme les méthodes d'échange, ces procédés sont très sensibles aux pressions temporelles. De ce fait, la rapidité de la procédure adoptée est primordiale si le degré de dynamisme est très élevé, d'où le recours à des implantations parallèles [28].

Choix de la longueur de l'horizon

Le choix de la longueur L de l'horizon fuyant est associé à différentes politiques de traitement de l'aspect dynamique. Dans ce qui suit, nous discutons deux cas extrêmes : L très petit et L infini. Un choix judicieux entre ces deux extrêmes dépend du problème étudié.

Cas où L est très petit

Lorsque L est très petit, une planification à très court terme est adoptée. Ceci permet d'atteindre des solutions très proches de l'optimum sur les données connues à l'intérieur de l'horizon. En revanche, cette stratégie engendre des solutions où le futur relativement proche est ignoré (approche de résolution myope). Un horizon court est tout à fait indiqué pour des problèmes où le degré de dynamisme est

élevé. En effet, une planification à moyen ou long terme s'avère inutile puisqu'il est très probable que de nouveaux événements surgiront entre temps et rendront le plan invalide.

Cas où L est très grand

Ici, le problème statique qu'on se propose de résoudre est plus riche puisqu'il tient compte de données couvrant un horizon étendu. Par conséquent, l'algorithme utilisé, limité par le temps d'exécution permis, doit se contenter de solutions de qualité généralement inférieure à celle produites dans le premier cas. Lorsque le degré de dynamisme est très élevé, cette méthode tient compte d'événements éloignés dans le temps qu'il n'est généralement pas utile de considérer immédiatement (coûts de calcul additionnel, changement de situation qui entraîne une non pertinence du plan...). En revanche, lorsque le degré de dynamisme du problème est très faible, il peut être acceptable de considérer un horizon étendu. C'est le cas, par exemple, des problèmes de routage et d'inventaire décrits précédemment.

Le principal reproche fait à l'égard des adaptations d'algorithmes statiques, c'est qu'elles ne tiennent compte que de l'information connue avec certitude, ignorant ainsi l'apport bénéfique d'informations de nature probabiliste portant sur le futur. C'est le sujet des deux prochaines sous-sections.

2.4.2 Méthodes stochastiques

Dans les problèmes de répartition de véhicules en temps réel, l'efficacité d'une méthode de résolution peut être améliorée si elle modélise mieux les aspects stochastiques du problème, et tient compte du futur. Aussi, les techniques stochastiques s'imposent-elles comme un moyen naturel d'arriver à ce but. Parmi ces techniques, nous citons la programmation stochastique et les modèles de décision

markoviens [51, 53, 56, 58]. Toutefois, plusieurs obstacles nuisent à la popularité de ces approches. Dans les processus de décisions markoviens, outre la difficulté de modéliser adéquatement le futur, la définition d'un état qui reflète la complexité de la réalité engendre des espaces de recherche de très grande taille. Cette difficulté est encore plus évidente si l'on considère que ces modèles sont souvent résolus par des algorithmes de programmation dynamique, ce qui empêche la résolution de problèmes de taille réaliste. Outre ce handicap, la programmation stochastique est une discipline relativement jeune qui n'a pas encore atteint sa pleine maturité.

2.4.3 Méthodes basées sur la théorie des files d'attente

Dans les problèmes à caractère très dynamique, le taux d'arrivée des requêtes est typiquement très élevé et la majorité des requêtes sont urgentes. Il en découle que le système devient vite congestionné et ne peut arriver à desservir toutes les requêtes dans des délais acceptables. Dès lors, la liste des requêtes pendantes s'allonge et les retards de service s'accentuent. Dans un tel contexte de saturation, les approches classiques de répartition et d'affectation basées sur des critères statiques sont voués à des résultats limités, voir erronés. Ceci est en accord avec les remarques de Psaraftis [67]. Dès lors, l'élaboration d'approches de résolution combinant à la fois des composantes de répartition et des résultats tirés de la théorie des files d'attente offre une voie très prometteuse. De nos jours, peu de travaux sont encore réalisés dans cette optique (voir cependant [7, 8, 87]).

2.5 Les problèmes de cueillette (ou livraison)

Les sujets que nous abordons ici s'appliquent aussi bien aux problèmes de cueillette (ou livraison) qu'aux problèmes mixtes avec cueillettes et livraisons.

Dans le premier cas, une requête est représentée par un seul point de service. Dans le second cas, il y a deux points de service avec une contrainte de précédence entre le point de cueillette et le point de livraison.

Bien qu'ils partagent des liens étroits avec les problèmes avec requêtes à un point, les problèmes avec requêtes à deux points posent des défis encore plus importants. Par exemple, l'existence d'algorithmes pour résoudre des problèmes à un seul véhicule incite souvent les chercheurs à ramener les problèmes multi-véhicules à ce dernier cas. Pour les problèmes avec requêtes à un point, ceci est souvent réalisé en regroupant les clients qui sont proches du point de vue temps et espace. Or, dans le cas de requêtes à deux points, une définition adéquate du concept de "requêtes proches" est un défi majeur. D'autre part, le développement d'heuristiques pour cette classe de problèmes est confronté au fait que la cueillette et la livraison d'une requête doivent se faire par le même véhicule et que la cueillette doit précéder la livraison.

Nous consacrons cette section à une revue des approches de résolution proposées pour les problèmes avec requêtes à un point. Dans la section suivante, nous examinerons les travaux relatifs aux problèmes avec requêtes à deux points.

2.5.1 Adaptation d'algorithmes statiques

Les problèmes de répartition de véhicules en temps réel font partie de la classe des problèmes NP-complets. Les algorithmes exacts développés jusqu'à présent demeurent donc impuissants devant les problèmes de grande taille rencontrés en pratique. Aussi, est-il intéressant de développer des approches heuristiques. Ces méthodes se sont souvent avérées très utiles et offrent un bon compromis entre la qualité de la solution et le temps d'exécution. Plusieurs heuristiques rapportées

dans la littérature sont basées sur des méthodes de recherche locale. Les plus répandues sont les méthodes d'insertion. Cet intérêt découle principalement de leur facilité d'implantation ainsi que leur temps d'exécution raisonnable qui convient parfaitement au caractère dynamique du problème. Le mécanisme général consiste à construire des routes initiales en se basant sur les requêtes connues à l'avance. Les demandes qui arrivent en temps réel sont ensuite insérées dans ces routes grâce à des règles simples mais rapides.

Les chercheurs sont aussi tentés par l'élaboration de méthodologies hybrides combinant des approches exactes et des heuristiques. Le but est de tirer profit des avantages de chaque catégorie : précision des méthodes exactes et souplesse des heuristiques lorsqu'il s'agit par exemple de contraintes très complexes. Dans la suite, nous décrivons ces différentes approches pour chacune des classes d'applications identifiées à la section 2.2.

Routage et inventaire

L'une des premières applications est celle de Bell et al. [4]. Des dépôts assurent l'approvisionnement d'un certain nombre de clients en produits gazeux. Chaque client doit recevoir une quantité conforme à ses besoins. Pour chaque client, il s'agit alors de décider de la quantité à livrer ainsi que de la date de la visite, de façon à déterminer les routes de chaque véhicule. Plusieurs contraintes sont présentes, dont les fenêtres de temps aux clients (dates de visite au plus tôt et au plus tard), les capacités des véhicules ainsi que la compatibilité entre un véhicule et les produits qu'il transporte. L'approche proposée consiste à appliquer l'algorithme statique de Fisher et al. [23] afin de trouver la répartition des véhicules pour les jours à venir (deux à cinq jours). Cet algorithme statique est basé sur une formulation en programmation linéaire mixte, à laquelle les auteurs appliquent une relaxation

lagrangienne et une méthode d'ajustement des multiplicateurs. Un module séparé permet au planificateur de mettre à jour interactivement les données qui surgissent en temps réel. Suite à cette mise à jour, la procédure est ré-appliquée à nouveau. Afin de décider de la date et de la quantité à livrer à chaque client, les auteurs font une estimation du taux de consommation de ce dernier en fonction de données historiques. En second lieu, tout en tenant compte de l'emplacement géographique des clients ainsi que des ressources disponibles, ils regroupent les clients devant être desservis par un même véhicule. Finalement, la résolution d'un programme linéaire mixte leur permet d'isoler un bon sous-ensemble de regroupements. Pour chacun de ces regroupements, ils spécifient la date de départ, les quantités à livrer aux clients ainsi que le véhicule à utiliser. La méthode est testée sur des problèmes avec 10 à 30 camions et 150 à 400 clients par dépôt par jour. Les auteurs rapportent que de tels problèmes engendrent un énorme programme linéaire mixte. Toutefois, la relaxation lagrangienne permet sa décomposition en problèmes de sac à dos, ce qui permet d'obtenir des solutions à moins de 2% de l'optimum.

Brown et al. [16] s'attaquent à un problème de répartition de véhicules pour une compagnie pétrolière. Il s'agit de satisfaire les demandes des clients à partir de différents dépôts où la demande peut être constituée de plusieurs produits. Chaque client a une fenêtre de temps et peut être desservi par un ou plusieurs camions. Les véhicules possèdent des capacités et des nombres de compartiments différents. L'objectif est de réaliser une répartition qui minimise les coûts de transport tout en équilibrant les tâches des camions et en assurant un bon service aux clients (soit à l'intérieur des fenêtres de temps).

Les auteurs utilisent un horizon fini, et lors de la répartition pour la période courante, ils considèrent exclusivement les requêtes dont les fenêtres de temps se si-

tuent à l'intérieur de cet horizon. Toute demande surgissant pendant le processus de répartition est considérée seulement lors de la prochaine période. L'algorithme statique employé est constitué de cinq étapes principales :

Étape 1 : En se basant sur l'analyse des données de la période courante, une affectation initiale des chargements aux différents camions est faite manuellement afin d'équilibrer les tâches de ces derniers.

Étape 2 : L'affectation réalisée à l'étape 1 est améliorée à l'aide d'échanges de chargements entre les camions.

Étape 3 : La meilleure séquence de cueillettes et de livraisons des chargements est déterminée pour chaque camion. Les auteurs utilisent l'algorithme de Graves et Whinston [34] afin de résoudre un problème de voyageur de commerce pour chaque véhicule.

Étape 4 : L'affectation des produits aux compartiments de chaque camion est déterminée de façon optimale. Les auteurs utilisent une méthode spécialisée d'énumération.

Étape 5 : Les chargements compatibles sont échangés entre deux camions afin d'améliorer la solution finale.

L'une des faiblesses de cette approche réside dans l'ignorance de toute requête qui apparaît à l'intérieur de la période courante. En effet, son traitement immédiat, plutôt que son report jusqu'à la période suivante, ne pourrait avoir que des effets bénéfiques sur la méthode. Toutefois, ceci est impossible, dû à la nature et aux temps de calcul de l'algorithme statique utilisé.

Dans le même contexte, Baush et al. [3] considèrent le cas d'une compagnie pétrolière qui dispose de dix dépôts dont chacun alimente un groupe de clients en différents produits. Des camions de capacités et de coûts différents assurent

le transport des requêtes (avec fenêtres de temps). Dans cette étude, les auteurs utilisent un horizon d'un jour. Ainsi, ils mettent à jour les données au début de la période courante avant d'exécuter leur algorithme statique afin de produire les routes des véhicules pour la journée. Le processus est composé de trois grandes étapes :

Étape 1 : Générer pour chaque type de véhicule un sous-ensemble des regroupements de requêtes qu'il peut satisfaire. Les auteurs utilisent une heuristique de balayage circulaire à cet effet [31]. À partir d'un rayon liant le dépôt à un client particulier, on se déplace dans le sens opposé à celui des aiguilles d'une montre jusqu'à ce qu'on atteigne la capacité maximale du véhicule.

Étape 2 : Ordonner les séquences obtenues en minimisant la distance ou le temps de parcours. Selon la taille de chaque séquence, les auteurs utilisent une énumération exacte ou une heuristique d'affectation quadratique.

Étape 3 : Spécifier pour chaque véhicule de chaque type, les dates de visite à ses clients, les arrêts et tous les détails du voyage. Les auteurs adoptent un modèle de partitionnement qu'ils résolvent par une méthode d'évaluation et séparation ("branch and bound") combinée à une méthode duale du simplexe.

Notons que les auteurs n'ont comparé les résultats obtenus qu'à ceux produits par des planificateurs humains, ce qui leur a permis des améliorations notables.

Services de courrier rapide

Hill et al. [35] adaptent une approche statique pour un système d'aide à la décision pour la gestion de courrier bancaire. Chaque jour, des véhicules transportent des chèques et d'autres items, de certaines succursales vers un centre de traitement. L'objectif est de minimiser la distance totale parcourue par les véhicules. Pour ce

faire, la construction des routes doit tenir compte des contraintes suivantes :

- Le temps d'ouverture de chaque succursale et la disponibilité de chaque véhicule.
- Nombre minimal et maximal d'arrêts par route.
- Temps maximal de parcours.
- Nombre minimal d'items requis au centre en tout temps.

L'horizon correspond ici à une journée. Toutes les quinze minutes, le nombre minimal d'items requis au centre ainsi que les temps de parcours entre les paires de succursales sont mis à jour. Afin de tenir compte de la congestion, les auteurs multiplient ces temps de parcours par des facteurs qui reflètent l'état de la circulation. L'algorithme statique qui est à la base de la méthode de résolution repose sur la construction séquentielle de routes, suivant le principe du "maximum delay". Selon ce principe, chaque route doit finir le plus tard possible tout en respectant la fenêtre de temps qui lui est associée. Le but est de recueillir le plus grand nombre d'items possibles sur chaque route. La fenêtre de temps de chaque véhicule est définie par le moment où ce véhicule devient disponible et celui où le nombre d'items au centre devient inférieur au minimum requis. La première étape consiste à insérer dans la route courante, parmi toutes les succursales non encore visitées, celle qui minimise le coût d'insertion (soit un ratio tenant compte des contraintes mentionnées plus haut). La deuxième étape s'amorce quand on ne peut plus insérer de succursales qui respectent l'intervalle spécifié pour la formation de la route. Des échanges sont alors appliqués dans l'espoir d'améliorer la solution. Les auteurs ont appliqué leur approche à un cas réel comprenant 140 succursales avec des résultats qui ont surpassé ceux obtenus par des planificateurs humains.

Gendreau et al. [25] proposent une autre approche de résolution pour un

problème de répartition avec fenêtres de temps souples aux clients et une fenêtre de temps rigide au dépôt. Les auteurs utilisent un horizon infini et résolvent un problème statique à chaque fois qu'un nouvel événement apparaît. Un tel événement peut être la fin de la tâche d'un véhicule ou l'arrivée d'une nouvelle requête. Une solution est un ensemble de routes planifiées commençant chacune par la destination courante d'un véhicule. L'objectif correspond à la somme de la distance totale parcourue par les véhicules et les retards de service aux clients. Les auteurs élaborent une heuristique basée sur la méthode de recherche avec tabous conçue par Glover [32, 33] à laquelle ils ajoutent une mémoire adaptative [77]. L'heuristique se résume comme suit :

- Construire I différentes solutions initiales avec une heuristique d'insertion stochastique.
- Appliquer la méthode de recherche avec tabous à chaque solution et conserver les routes obtenues dans la mémoire adaptative.
- Tant qu'un critère d'arrêt n'est pas satisfait, faire :
  - Construire une solution initiale à partir des routes conservées dans la mémoire adaptative.
  - Décomposer le problème en sous-problèmes en utilisant une procédure de décomposition.
  - Appliquer la méthode de recherche avec tabous à chaque sous-problème.
  - Mettre les meilleures routes obtenues dans la mémoire adaptive.
- Optimiser individuellement chaque route de la meilleure solution produite.

Les principaux éléments de cet algorithme sont décrits ci-dessous.

La mémoire adaptive

Les meilleures solutions rencontrées par l'algorithme sont gardées en mémoire

adaptive afin de générer de nouveaux points de départ pour la méthode de recherche avec tabous. Le processus de construction d'une solution consiste à combiner des routes pour former une solution partielle ou complète, en favorisant les meilleures routes en mémoire adaptive. Les requêtes non desservies à la fin du processus sont insérées par l'heuristique de Solomon [83].

Décomposition/Reconstitution

Chaque solution est décomposée en D sous-ensembles de routes et chaque sous-ensemble est traité par une méthode de recherche avec tabous différente. Les solutions obtenues sont ensuite regroupées pour constituer la solution finale. Ce processus est répété un certain nombre de fois au cours de la recherche et, à la fin, les routes de la meilleure solution rencontrée sont stockées dans la mémoire adaptive.

La méthode de recherche avec tabous

La méthode de recherche avec tabous est très semblable à un algorithme de descente permettant le passage d'une solution courante à une autre solution dans le voisinage de cette solution courante. Le rôle de la méthode de recherche avec tabous est de permettre à cet algorithme de descente d'échapper aux minima locaux. Ainsi, quand aucune transition de descente n'est possible, l'algorithme permet la transition de plus petite montée. Par la suite, on risque de revenir à la solution précédente (qui était meilleure). Pour réduire le danger de cyclage, il est nécessaire d'utiliser des listes, appelées listes tabou, qui contiennent des transitions interdites. On y ajoute les dernières transitions effectuées, et on en retire les plus anciennes. Le caractère tabou d'une transition peut être levé si elle permet une amélioration de la meilleure solution connue. Vu que l'on permet des transitions détériorant l'objectif, on arrête typiquement la procédure dès qu'on dépasse un nombre maxi-

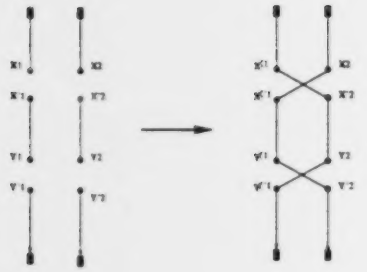

FIG. 2.1 – L'échange CROSS.

mal d'itérations consécutives sans améliorer la solution.

Voisinage

Le voisinage de la solution courante est généré par l'échange "CROSS" qui consiste à échanger deux segments de longueur variable entre deux routes différentes ou à l'intérieur d'une même route. La figure 2.2 illustre cette opération pour deux routes où les segments $X_1' - Y_1$ et $X_2' - Y_2$ sont échangés.

Optimisation finale

Chaque route de la solution finale est optimisée par une heuristique du voyageur de commerce avec fenêtres de temps, conçue par Gendreau et al. [26].

Les décisions en temps réel

Les auteurs considèrent deux types d'événements :

1- Lorsqu'un véhicule a complété le service à un client, sa prochaine destination est déterminée par la meilleure solution en mémoire adaptive. Toutes les solutions en mémoire adaptative sont alors mises à jour en conséquence.

2- L'apparition d'une nouvelle requête entraîne son insertion dans toutes les solutions en mémoire adaptive, à l'endroit qui minimise la somme du détour et des retards additionnels aux clients. Après élimination des solutions où une telle insertion est impossible, la meilleure solution en mémoire adaptive est améliorée par

une heuristique de descente locale basée sur l'échange "CROSS".

Une fois ces mises à jour accomplies, une nouvelle solution initiale est créée pour la recherche avec tabous en combinant des routes stockées dans la mémoire adaptative.

Parallélisme

Afin d'augmenter la quantité de travail effectué par la méthode de recherche avec tabous entre deux événements, une implantation parallèle sur un réseau de stations de travail a été réalisée. Cette parallélisation est réalisée à deux niveaux :

1- Plusieurs trajectoires de recherche s'exécutent en parallèle et communiquent leurs résultats à la mémoire adaptative.

2- À l'intérieur d'une même trajectoire de recherche, le processus de décomposition d'un problème en sous-problèmes est également parallélisé.

Les auteurs comparent leur approche à différentes méthodes heuristiques comme l'insertion simple et l'insertion combinée à des heuristiques de descente. Les résultats favorisent la méthode proposée.

Services d'urgence

Trudeau et al. [90] s'intéressent au problème de planification et de gestion d'un système d'urgence dans le domaine de la santé. Le travail consiste à répartir les unités et leurs équipes afin de satisfaire des demandes ayant lieu en temps réel. Plusieurs facteurs, dont le lieu et la date de chaque requête, sont de nature stochastique. En outre, les auteurs considèrent le problème de relocalisation des unités libres à certains points de relève fixés à priori. Le processus doit tenir compte de certaines contraintes comme les capacités des ressources, les heures de travail admissibles pour le personnel, etc. Les auteurs proposent un modèle de simulation pour évaluer différentes stratégies. Ce modèle repose sur deux modules de décision.

Le premier concerne la phase de planification. Ainsi, pour un jour typique de la semaine, ils déterminent le nombre d'unités en service à chaque instant, en se basant sur des modèles de prévision. Ils spécifient aussi l'affectation de ces véhicules et les horaires de leurs équipes à l'aide d'un algorithme statique de programmation en nombres entiers [12]. Le deuxième module concerne les décisions à prendre en temps réel. Ainsi, lorsqu'une nouvelle requête apparaît, dépendant de sa priorité, elle est satisfaite ou remise à plus tard si le nombre de ressources disponibles est sous un certain seuil. La requête retenue est confiée au véhicule le plus près parmi tous les véhicules disponibles. Celui-ci, selon la stratégie employée, la transporte au centre hospitalier le plus près, ou celui ayant le taux d'occupation le plus bas. Lorsque le véhicule termine sa tâche, certains des véhicules libres sont relocalisés à un point particulier parmi un ensemble de points de relève fixés à priori.

Dans le même contexte, Gendreau et al. [30] se sont récemment intéressés à la question du repositionnement des véhicules en temps réel. L'approche de résolution proposée est une extension de l'algorithme développé antérieurement par les auteurs pour la version statique du problème [29]. Considérant deux types d'appels, urgents ou non, le but est de maximiser le nombre de points d'appels couverts par deux véhicules tout en respectant les contraintes de recouvrement suivantes : 90 % des temps d'arrivée sur les lieux pour des appels urgents doivent être inférieurs à 5 minutes et 100 % des temps d'arrivée sur les lieux pour des appels urgents doivent être inférieurs à 10 minutes. L'algorithme statique utilisé est une heuristique parallèle basée sur la recherche avec tabous [32, 33]. Afin de construire le voisinage de la solution courante, les auteurs définissent la notion de "chaînes d'aspiration". Une telle chaîne est construite en déplaçant une ambulance donnée de sa position courante vers une autre position non saturée (i.e. une position où le nombre maxi-

mal d'ambulances en attente est non encore atteint). Les ambulances sont ensuite deplacées de façon à respecter les contraintes de recouvrement. Des simulations produites à partir de données réelles confirment la pertinence de l'approche proposée.

Services de réparation

Dans sa thèse, Larsen [44] considère une variante du problème du réparateur introduit par Bertsimas et Van Ryzin [7, 8]. Dans cette variante, un certain nombre de requêtes sont connues à l'avance et l'objectif est de minimiser la distance totale parcourue. L'auteur fait appel à la simulation pour évaluer la performance des stratégies locales analysées par Bertsimas et Van Ryzin [7, 8]. Ces simulations reproduisent différents degrés de dynamisme. Les résultats montrent que pour toutes les stratégies considérées, la distance totale parcourue croît linéairement en fonction du degré de dynamisme du problème. En outre, dans un système congestionné où le taux d'arrivée des requêtes est très élevé, la stratégie du "voisin le plus proche" fait mieux que toutes les autres stratégies considérées.

Madsen et al. [47] considèrent une autre application du problème de réparation à domicile. Dans cette application, les auteurs considèrent le cas d'une compagnie spécialisée dans les produits gazeux industriels. Dans ce problème, il s'agit d'identifier une stratégie optimale de planification des routes sur un horizon d'une semaine. Les fenêtres de temps sont fixées par la compagnie et correspondent à la période durant laquelle le client sera visité. L'objectif est de minimiser la distance totale parcourue. Lorsqu'une nouvelle requête arrive, celle-ci est insérée à la position minimisant le détour parmi toutes les positions possibles dans un sous-ensemble de routes préalablement choisies. Finalement, une fenêtre de temps est associée à la requête en prenant soin d'équilibrer les tâches affectées aux différentes

périodes de travail.

Notons que plusieurs travaux présentés dans cette sous-section sont motivés par des projets commerciaux. Il ressort que les stratégies adoptées sont souvent assez simples. Par ailleurs, il est difficile de comparer ces méthodologies entre elles puisque les objectifs divergent et que les environnements de développement sont loin d'être uniformes. Finalement, remarquons que mis à part les ambulances et les services de courrier rapide, les problèmes considérés sont des problèmes à faible degré de dynamisme. Ceci explique le bon fonctionnement d'adaptations simples d'algorithmes statiques. Ceci est en accord avec les observations de Lund [45].

2.5.2 Méthodes stochastiques

Le principal reproche fait à l'égard des adaptations d'algorithmes statiques, c'est qu'elles ne tiennent compte que d'informations connues avec certitude, ignorant ainsi l'apport bénéfique d'informations sur le futur. Les techniques stochastiques s'imposent comme un moyen naturel de remédier à cette faiblesse.

Routage et inventaire

Minkoff [51] considère un problème qui consiste à utiliser une flotte de véhicules afin d'alimenter périodiquement les stocks d'un ensemble de clients ayant des demandes stochastiques, où $q_j(x)$ est la probabilité que le client j consomme x unités de ressources dans une période donnée. Lorsque le véhicule est disponible, le planificateur se base sur l'inventaire actuel de chaque client afin de choisir un itinéraire pour ce véhicule. Soulignons qu'un itinéraire est composé d'un ensemble de clients à visiter par le même véhicule, ainsi que des quantités de ressources à leur fournir. L'obtention d'un ensemble initial d'itinéraires ainsi que la construction des routes ne sont pas considérées par les auteurs. Le but est simplement de trouver une

affectation d'itinéraires prédéfinis aux véhicules de sorte que le coût total (c-à-d. somme des coûts de transport, de stockage et de consommation) soit minimal. Les auteurs modélisent ce problème comme un processus de décision markovien, où l'état du système correspond aux niveaux d'inventaires des clients au début de chaque période de répartition. L'approche de résolution est une heuristique qui se résume comme suit :

1- Résoudre pour chaque client un problème de décision markovien de petite taille. La solution de chacun de ces problèmes exprime la quantité de ressources à livrer au client associé, en fonction du niveau d'inventaire en début de période.

2- En utilisant les résultats de l'étape 1, définir un problème de programmation en nombres entiers pour obtenir l'affectation des itinéraires. Les auteurs résolvent ce problème par la méthode de séparation et évaluation.

Notons que la limitation majeure de ce travail provient des hypothèses émises afin de simplifier le modèle général (p.e. on suppose un nombre illimité de véhicules). Ces simplifications préviennent l'application de l'approche proposée à des problèmes réels.

Services de courrier rapide

Larsen [44] étudie un problème dynamique de voyageur de commerce avec fenêtres de temps. Les requêtes proviennent de n différentes sous-régions selon un processus de Poisson avec paramètres d'intensité λ_i, $i = 1, ..., n$. Chaque sous région possède un ou plusieurs points de repos pour les véhicules. En outre, toutes les requêtes doivent être servies même si cela induit des retards au dépôt. L'objectif est de minimiser une somme pondérée du temps de parcours et du retard total. Lorsqu'une nouvelle requête apparaît, le tour courant est réoptimisé en utilisant une procédure

d'échanges. L'auteur aborde également aussi la question du repositionnement des véhicules libres en prévision des demandes futures. Pour ce faire, il exploite une connaissance probabiliste des arrivées des requêtes. Ainsi, lorsqu'un véhicule a fini de servir son client courant, l'algorithme détermine le meilleur point de repos pour ce véhicule (si c'est possible) et la date de début de service chez son prochain client. Les trois règles suivantes sont utilisées à cet effet : **(i)** le véhicule attend au point de repos le plus proche de sa position courante ; **(ii)** le véhicule attend au point de repos le plus occupé (c-à-d., celui qui est situé dans la sous-région correspondant à la plus grande valeur λ_i) ; **(iii)** le véhicule attend au point de repos situé dans la sous-région correspondant au plus grand nombre espéré de clients. Toutefois, si la probabilité qu'au moins une requête apparaisse dans la sous-région choisie est inférieure à un seuil donné, le véhicule attend à sa position courante. Des simulations comparent ces différentes stratégies de relocalisation incluant la stratégie de "non repos" où le véhicule attend systématiquement à sa position courante. Cependant, les résultats numériques ne permettent pas d'identifier clairement une stratégie dominante.

2.5.3 Les méthodes basées sur la théorie des files d'attente

Dans un environnement très congestionné comme les systèmes publics d'urgence, les méthodes classiques basées sur des critères statiques ne sont pas appropriées. En revanche, le jumelage de techniques inspirées des résultats de la théorie des files d'attente aux approches "conventionnelles" de répartition semble très prometteur.

Services de réparation

Bertsimas et Van Ryzin [7] proposent et analysent le problème dynamique du

réparateur. Les demandes de service arrivent suivant un processus de Poisson d'intensité λ et sont servies par un seul véhicule. Elles sont indépendamment et uniformément distribuées dans une région A euclidienne. Le temps de service à chaque client suit une distribution de probabilités quelconque. L'objectif est d'identifier une politique optimale qui minimise l'espérance du temps que chaque requête passe dans le système. En se basant sur des résultats de probabilités géométriques, de théorie des files d'attente et d'optimisation combinatoire, les auteurs présentent une étude analytique du problème dans les deux cas particuliers suivants : trafic lourd et trafic léger. Dans un premier temps, ils établissent des bornes inférieures pour le temps espéré passé dans le système. Par la suite, ils proposent et analysent des politiques simples (p.e. premier arrivé premier servi (FIFO)) et comparent leur performance aux bornes inférieures qu'ils ont établies.

En 1993, les auteurs tentent de généraliser leurs résultats au cas d'une flotte homogène de m véhicules dont chacun peut visiter au maximum q clients avant de revenir à son dépôt [8]. Ce problème se ramène au cas d'un seul véhicule en subdivisant la région A en sous-régions, sous des conditions de symétrie. Ensuite, ils utilisent les résultats obtenus pour leur première étude afin de concevoir des politiques qui se rapprochent de l'optimum en cas de trafic lourd ou léger.

Ce travail se range parmi les études théoriques dont nous avons présenté les limitations au début de ce chapitre. Par ailleurs, la subdivision de la région A en sous-régions n'est pas une question simple. En revanche, comme le soulignent Bertsimas et Simchi-Levi [9], ce genre d'étude permet de mieux caractériser la structure des solutions, offrant ainsi une bonne base pour le développement d'algorithmes plus efficaces.

2.6 Les problèmes mixtes de cueillette et livraison

Les problèmes de répartition de véhicules pour les requêtes à deux points, aussi nommés problèmes de cueillette et livraison en temps réel, possèdent un champ d'application très vaste. Citons entre autres le transport adapté, le transport d'urgence, le courrier rapide et le transport de marchandises. Nous présentons les méthodes de résolution proposées dans la suite.

2.6.1 Adaptation d'algorithmes statiques

Transport sur demande

Psaraftis [61] adapte au contexte dynamique une étude qu'il a effectuée pour un problème statique de transport sur demande sans fenêtres de temps et avec un seul véhicule de capacité finie. Dans la version statique, chaque client requiert un service le plus tôt possible. L'objectif est alors une somme pondérée du délai de service et de l'"inconvénience" aux clients. Cette "inconvénience" correspond au délai de service au point de cueillette et au temps de parcours jusqu'à la livraison. Dans la version dynamique, les requêtes arrivent en temps réel. Lorsqu'une nouvelle demande arrive, l'auteur élimine toutes les requêtes déjà desservies et ajoute la nouvelle requête à celles non encore satisfaites. Par ailleurs, après avoir mis à jour la position du véhicule, il ramène à cette position les points de cueillette de tous les clients dont la requête a été recueillie mais non encore livrée. Enfin, il modifie la route planifiée en utilisant l'approche conçue pour le cas statique, soit un algorithme exact (donc de complexité exponentielle) de programmation dynamique. Un vecteur état $(L, k_1, k_2, k_3, ...)$ est défini comme suit :

- L est la position courante du véhicule ($L = 0$ au dépôt, $L = i$ au point de prise du client i et $L = i + n$ au point de livraison du client i).
- k_i est le statut du client i ($k_i = 3$ si i n'est pas encore recueilli, $k_i = 2$ si i est recueilli mais non encore livré, et $k_i = 1$ si i est déjà livré).

Il importe de noter qu'en vue d'éviter de remettre constamment le service à un client situé dans une zone périphérique, l'auteur impose des contraintes de priorité définies de la façon suivante :

$| (i - p_i) | \leq M$ et $| (i - d_i) | \leq M'$

où i dénote l'ordre de réception de la requête,

p_i dénote l'ordre de la requête i dans la séquence des points de cueillette,

d_i dénote l'ordre de la requête i dans la séquence des points de livraison,

et M, M' sont des constantes.

Étant donné le temps d'exécution de la procédure statique croît exponentiellement avec le nombre de requêtes, un horizon très court doit être choisi afin que celle-ci n'opère que sur un nombre de clients réduit.

Plus tard, le même auteur [63] adapte l'approche évoquée au cas où chaque point de cueillette et de livraison possède une fenêtre de temps. Les fenêtres de temps sont respectées en éliminant les états non réalisables.

Psaraftis [65] développe aussi un algorithme nommé MORSS pour la répartition dynamique de navires militaires. Outre les fenêtres de temps, les capacités des navires et celles des ports sont prises en considération. L'auteur utilise un horizon de longueur L. Ainsi, au temps t_k correspondant à l'itération k, la procédure s'occupe uniquement des requêtes connues ayant des temps de prise au plus tôt e_i^+ situé entre t_k et $t_k + L$. Elle affecte alors ces requêtes aux navires admissibles. Toutefois, cette affectation n'est permanente que pour les chargements i où $e_i^+ \in [t_k, t_k + aL]$,

$0 \leq a \leq 1$. Pour les autres, cette affectation n'est que temporaire, permettant ainsi de s'ajuster à d'éventuels changements. Après les mises à jour nécessaires, la procédure passe au temps t_{k+1} qui se produit au moment où un nouvel événement surgit ou alors au temps $e_i^{+*} = Min\{e_i^+ / i$ est une requête non encore servie$\}$.
L'affectation des requêtes aux navires se fait en deux étapes. Au cours de la première étape, l'auteur calcule le coût d'affectation U_{ij} de la requête i au navire j. Ce coût est une fonction complexe qui mesure l'effet de cette affectation sur la date de livraison de i et celle de chaque requête déjà affectée au navire j, de l'efficacité d'utilisation du navire j, ainsi que des ressources des différents ports. Dans la deuxième étape, il optimise le problème d'affectation suivant :

$$\min \sum_i \sum_j U_{ij} x_{ij} \tag{2.1}$$

sujet à :

$$\sum_i x_{ij} \leq K, \quad \forall j \tag{2.2}$$

$$\sum_j x_{ij} \leq 1, \quad \forall i \tag{2.3}$$

$$x_{ij} = 0 \; ou \; 1, \quad \forall i, \quad \forall j \tag{2.4}$$

avec K un entier fixé qui représente le nombre maximal de requêtes que l'on peut servir par un même navire et x_{ij} une variable binaire qui prend la valeur 1 si la requête i est affectée au navire j, et 0 sinon.

Dans un contexte de transport sur demande, Wilson [94] propose une approche qui se décrit comme suit. Lorsqu'une nouvelle requête apparaît, il l'insère à la meilleure place parmi toutes les routes existantes. Cette recherche correspond à un arbre à trois niveaux : le premier niveau contient tous les véhicules disponibles,

le second contient toutes les insertions possibles pour le point de collecte de la requête, et le troisième contient toutes les insertions possibles pour le point de livraison. L'auteur adopte un algorithme de séparation et évaluation pour explorer l'arbre de recherche. Le coût d'insertion contient deux composantes : la première est une fonction de pénalité quadratique sur les temps de prise et de parcours ; la seconde vise à équilibrer la longueur des tournées des différents véhicules.

L'auteur propose deux extensions à son travail :

- La première extension consiste à réinsérer des requêtes déjà affectées à des véhicules mais non encore servies au bout de vingt minutes.
- dans la seconde extension, on conserve les deux meilleures insertions possibles au lieu d'une seule.

Madsen et al. [46] adaptent l'approche de Jaw et al. [41] initialement développée pour un problème statique de transport sur demande avec fenêtres de temps, à un cas dynamique de transport adapté où les véhicules ont des capacités multiples (p.e. un véhicule peut transporter un lit ou quatre sièges). Lorsqu'une nouvelle requête apparaît, elle est ajoutée dans une liste où les requêtes non encore satisfaites sont classées par ordre de priorité d'insertion (p.e. les requêtes qui ont des fenêtres de temps étroites sont prioritaires). Une pause peut également être insérée dans une route pour modéliser une panne ou une période d'entretien d'un véhicule. Chaque requête est insérée à la meilleure place réalisable dans l'ensemble des routes. Si aucune insertion n'est possible, la requête est allouée à un véhicule plus coûteux (p.e. un taxi). Le coût d'insertion est une somme pondérée du temps de parcours et du temps d'attente, de la déviation par rapport au temps de livraison désiré ainsi que des capacités non exploitées des véhicules. Les auteurs se fixent un temps maximal pour desservir les requêtes et toute requête non recueillie à la fin de cette

période est remise dans la liste des requêtes non servies. Les auteurs ont testé leur méthode sur des problèmes réels avec 300 requêtes et 24 véhicules, où l'insertion d'une nouvelle demande se fait en moins d'une seconde.

Rappelons que le principal reproche fait à l'égard des méthodes d'insertion est leur myopie. En effet, une suite d'insertions risque de produire une solution de pauvre qualité, alors qu'une réorganisation complète de la route permettra souvent d'obtenir une bien meilleure solution. Afin de remédier à cette situation, plusieurs techniques de réoptimisation sont proposées dans la littérature.

Roy et al. [79] exploitent cette idée dans la résolution d'un problème de transport adapté avec fenêtres de temps. Les requêtes non encore servies sont insérées une à une, chacune à la position qui minimise le détour spatial et temporel parmi toutes les positions d'insertion possibles dans toutes les routes. Suite à l'insertion de toutes les requêtes dans l'horizon courant, ils effectuent des échanges de requêtes entre les routes en vue de réduire l'inconvénient. Ici, l'inconvénient d'une route est une somme pondérée de la déviation par rapport aux temps désirés de cueillette et livraison, et de la différence entre le temps de parcours réalisé et le temps de parcours direct entre le point de cueillette et le point de livraison de chaque requête.

Rivard [76] termine la phase d'insertion par la procédure de réoptimisation suivante :

Pour chaque parcours I, pour chaque requête $j \in I$, faire :

- Retirer j de I et recalculer les temps de cueillette et de livraison pour les autres requêtes de I.
- Choisir les cinq parcours où l'insertion de j introduit les plus petites déviations par rapport aux temps courants de cueillette et de livraison de leurs clients.

– Insérer j dans le meilleur parcours (selon la fonction objectif).

L'auteur suggère de n'exécuter cette réoptimisation qu'à des intervalles fixes dans la journée (p.e. toutes les demi-heures).

Dial [20] propose une approche hybride composée d'une phase d'insertion et d'une phase de réoptimisation. Ainsi, lorsqu'une nouvelle requête arrive, il l'affecte au véhicule qui correspond à l'insertion de moindre coût. Par la suite, l'auteur propose d'adopter l'approche exacte de Psaraftis [63] afin d'optimiser la route de chaque véhicule. Cette procédure étant exponentielle, il propose d'utiliser un horizon fini de très courte durée de façon à réduire le nombre de requêtes à considérer au cours de la résolution de chaque problème statique. Notons toutefois que l'auteur n'a fourni aucun résultat expérimental.

Camionnage

Powell et al. [54] présentent une modélisation par réseau. Ils partagent la région étudiée en R sous-régions et divisent l'horizon de longueur P en $(P+1)$ jours. L'ensemble des noeuds est R x $\{0, 1, ...P\}$. L'ensemble des arcs se divise en deux types :

- Des arcs déterministes qui peuvent représenter des mouvements avec charge ou à vide entre deux noeuds. Ces arcs sont connus au temps $t = 0$.
- Des arcs stochastiques dont chacun est dirigé vers un noeud (r, t) qui représente l'impact sur le futur du maintien du véhicule en attente dans une région r au temps t.

Les arcs avec charge possèdent une capacité de 1 et un coût associé à la prise en charge du client. Les autres arcs ont chacun une capacité infinie et un coût nul. Les véhicules entrent dans le réseau à la date et dans la région où ils sont disponibles. Le problème consiste alors à déterminer un flot à valeur maximale dans ce réseau.

Pour ce faire, les auteurs utilisent un algorithme exact basé sur une adaptation du simplexe. Cette procédure est exécutée quatre fois par jour après les mises à jour nécessaires. Notons que l'approche proposée n'est sans doute pas très efficace dans le cas où les informations changent très fréquemment car le réseau est typiquement de grande taille.

Powell et al. [60] s'interrogent sur la pertinence d'adopter des approches optimales globales au lieu de procédures voraces locales dans un environnement dynamique stochastique où l'on ignore toute information probabiliste sur le futur. Les auteurs considèrent trois sources d'incertitude : les arrivées des requêtes, les temps de parcours ainsi que la décision du planificateur d'utiliser ou de rejeter la solution suggérée par l'algorithme. Les résultats numériques montrent que dans un environnement où le degré d'incertitude est relativement élevé et où l'on ignore toute information probabiliste sur le futur, l'effort fourni en vue d'obtenir des solutions optimales n'est pas justifié.

Dans le même contexte, Regan et al. [72, 73, 74, 75] présentent différentes règles locales pour l'affectation des requêtes aux véhicules en temps réel. Ainsi, lorsqu'une nouvelle requête arrive, les auteurs examinent la possibilité de l'insérer tout simplement à la fin de la liste des requêtes pendantes ou de faire suivre cette insertion par un réordonnencement. Les auteurs s'intéressent aussi à la diversion qui consiste à détourner un véhicule de sa destination courante, alors qu'il est déjà en mouvement, afin de lui permettre de servir une requête nouvellement apparue à proximité de sa position courante. Des simulations ont démontré l'efficacité de ces opérations locales dans un environnement dynamique stochastique.

Dans un travail ultérieur, Yang et al. [91] comparent les solutions obtenues par des opérations locales à celles obtenues par un algorithme exact basé sur un modèle

mathématique d'affectation, et qui est résolu pour chaque nouvelle requête. Des simulations effectuées sur des problèmes avec 500 requêtes et 4 véhicules démontrent que l'algorithme exact aboutit à des solutions de meilleure qualité que celles obtenues par des méthodes locales. Toutefois, le temps d'exécution élevé requis pour l'exécution ce cet algorithme limite son application à des problèmes de petite taille. D'autre part, plusieurs des opérations locales considérées aboutissent à des solutions quasi-optimales dans des délais d'exécution plus courts.

Dans le problème considéré dans Fisher et al. [24], les véhicules peuvent travailler deux périodes par jour et les requêtes doivent être servies durant l'une ou l'autre de ces deux périodes. L'objectif est de minimiser la distance totale parcourue, en respectant la durée maximale de disponibilité des chauffeurs. Les auteurs utilisent un horizon d'un jour. Ils proposent une heuristique basée sur une approximation du problème par un problème de transport. Les variables de décision correspondent aux flots des camions vides allant du point de livraison d'un client aux points de cueillette des autres clients. La solution de ce problème est un ensemble de cycles correspondant aux requêtes desservies par un même véhicule. Toutefois, un tel cycle peut ne pas respecter la durée de disponibilité du chauffeur, ou encore, ne pas contenir le dépôt. Les auteurs utilisent une heuristique vorace afin de combiner ou de décomposer des cycles, et ajouter un dépôt.

Powell et al. [59] présentent deux heuristiques pour résoudre le problème statique d'affectation des camions aux requêtes, qui est résolu pour l'horizon courant. Ces deux heuristiques font partie de la famille des procédures de génération de colonnes. Toutefois, contrairement à ces dernières, l'ajustement dual et la génération de colonnes sont intégrés dans une seule procédure au lieu de suivre un schéma de type "problème maître/sous-problème". Des tests numériques démontrent que les

heuristiques proposées conviennent à un environnement dynamique. En effet, elles génèrent des solutions de bonne qualité dans des délais de calcul raisonnables.

Mahmassani et al. [48] considèrent un problème de camionnage sans consolidation. Les auteurs proposent une approche qui affecte les requêtes aux véhicules en deux étapes. Dans la première étape, les opérations locales et rapides proposées par Regan et al. [72, 73, 74, 75] ainsi que par Yang et al. [91] sont utilisées pour insérer une requête nouvellement apparue. Le but est d'aviser le client assez rapidement de l'acceptation ou du rejet de sa requête. Dans la seconde étape, l'algorithme exact proposé par Yang et al. [91] est utilisé pour réoptimiser la séquence des requêtes pendantes en attendant la prochaine requête. Tel que mentionné précédemment, cet algorithme requiert un temps d'exécution élevé. Les auteurs proposent deux stratégies pour palier à ce problème. Dans la première, seules des sous-séquences de requêtes pendantes sont considérées dans la phase de réoptimisation. Les requêtes retenues sont celles dont la date de cueillette est antérieure à un seuil temporel fixé. La seconde stratégie est réalisée en deux étapes. Dans la première étape, on fusionne toute paire de requêtes consécutives dont la distance est inférieure à un seuil donné. Dans la seconde étape, les séquences ainsi réduites sont réoptimisées à l'aide de l'approche exacte de Yang et al. [91]. Les résultats numériques montrent que la première stratégie améliore davantage la qualité de la solution, mais la seconde est plus rapide.

Services de courrier rapide

Rousseau et Roy [78] proposent un système d'aide à la décision pour un problème de courrier rapide dans un contexte urbain. Ils commencent par construire des routes pour servir les requêtes connues à l'avance. Ce processus se fait en rassem-

blant les clients dans des groupes avant de construire les horaires des véhicules qui servent chacun de ces groupes de façon à ce que les fenêtres de temps et les capacités des véhicules soient respectées. Lorsqu'une nouvelle requête surgit, ils l'insèrent au meilleur endroit à l'intérieur d'une route. Le choix du véhicule est basé sur différents critères comme :

- Choisir le véhicule le plus près du client.
- Choisir le véhicule pour lequel l'insertion du client provoque le plus petit détour.
- Choisir le véhicule ayant la plus grande capacité résiduelle.

Gendreau et al. [27] adaptent l'heuristique de recherche avec tabous développée dans un contexte de courrier rapide international (problème avec requêtes à un point) [28] au cas du courrier rapide local (problème avec requêtes à deux points). Pour ce faire, les auteurs changent la définition du voisinage de la solution courante. En effet, l'échange "CROSS" utilisé pour les requêtes à un point risque de placer les points de cueillette et de livraison d'une même requête dans des routes différentes. Afin de résoudre cette difficulté, les auteurs utilisent des chaînes d'éjection [89]. Ainsi, une requête dans une route donnée, avec son point de cueillette et de livraison, éjecte une autre requête appartenant à une seconde route, qui à son tour éjecte une requête appartenant à une troisième route, et ainsi de suite. Ce processus se termine par l'insertion de la dernière requête éjectée dans la première route de la chaîne ou dans toute autre route non encore visitée. La chaîne d'éjection peut être de longueur arbitraire mais doit être élémentaire (c-à-d., on ne peut pas revenir en arrière dans la chaîne pour éjecter une deuxième requête d'une route déjà visitée). Notons que le voisinage généré par les chaînes d'éjection peut être de très grande taille. Pour cette raison, les auteurs proposent un algorithme de plus court

chemin qui identifie la meilleure chaîne d'éjection dans un sous-ensemble réduit du voisinage complet.

Hooker et Natraj [36] utilisent aussi une approche basée sur la métaheuristique de recherche avec tabous pour résoudre un problème avec requêtes à deux points et fenêtres de temps. Chaque véhicule peut aussi avoir une origine, une destination et des fenêtres de temps. D'autres contraintes plus complexes peuvent s'ajouter, et une requête peut être satisfaite par plusieurs voyages. Afin de construire le voisinage de la solution courante, ils définissent la notion de "chaînes O-D généralisées". Une telle chaîne est une séquence ordonnée de paires $((j_1, q_1), (j_2, q_2), ..., (j_m, q_m))$, où j_i dénote un arrêt d'un véhicule p (cueillette ou livraison), et q_i représente la quantité recueillie ou livrée au point i selon que $q_i > 0$ ou $q_i < 0$. En outre la chaîne $(j_1, j_2, .., j_m)$ doit respecter les contraintes de précédence et chaque arc (j_i, j_k) doit vérifier la condition suivante :

$$E(j_i) + T_p(j_i, j_k) \leq L(j_k),$$

où

- $E(j_i)$ dénote la date de prise ou de livraison au plus tôt de la requête j_i.
- $L(j_k)$ dénote le temps de prise ou de livraison au plus tard de la requête j_k.
- $T_p(j_i, j_k)$ est le temps minimal requis par le véhicule p pour aller de j_i à j_k.

Une solution est représentée par une partition du graphe schématisant les cueillettes et les livraisons par un ensemble de chaînes O-D généralisées. Chacune d'elles forme les arrêts visités par un même véhicule et respecte la capacité de ce dernier. Afin de générer le voisinage d'une solution, dans un premier temps, les auteurs combinent les chaînes O-D entre elles de façon à obtenir de nouvelles chaînes qui respectent les contraintes de précédence, les fenêtres de temps ainsi que les capacités des véhicules. Cette combinaison se fait en mettant bout à bout deux sous-chaînes

de deux chaînes O-D données. Dans un second temps, ils insèrent les requêtes restantes en maintenant la réalisabilité.

2.6.2 Les approches stochastiques

Bien qu'elles représentent une réponse naturelle aux problèmes de répartition dynamique de véhicules, plusieurs limitations s'opposent à l'adoption d'approches stochastiques sur une plus grande échelle (voir la sous-section 2.3.2).

Camionnage

Powell [53] présente trois approches stochastiques dans un contexte de camionnage sans consolidation. Il commence par présenter un modèle de réseau dynamique basé sur un graphe espace-temps où chaque noeud correspond à une demande connue ou prévue dans une région donnée, à une date donnée. Les arcs se divisent en trois catégories : des arcs qui traduisent des déplacements connus de camions vides, d'autres qui modélisent des déplacements connus de camions chargés et d'autres (fictifs) qui représentent l'impact de la période courante p sur le futur. Chaque noeud (i, p) où i est l'indice d'une région, émet un arc vers un noeud fictif dont le coût $p_i(p+1)$ est une estimation du profit de localiser un camion additionnel dans la région i à la période $(p+1)$. Toutefois, ce modèle peut suggérer de servir une requête prévue même si elle ne s'est vraiment pas concrétisée. Afin de palier à ce problème, l'auteur propose un modèle stochastique dynamique où la demande $D_{ij}(t)$ entre les noeuds i et j à la période t est une variable aléatoire. De ce fait, le flot $x_{ij}(t)$ des camions qui se déplacent chargés, ainsi que le flot $y_{ij}(t)$ des camions qui se déplacent à vide entre i et j à la période courante t sont aussi des variables aléatoires. Ce modèle est présenté comme un problème de programmation stochastique à deux niveaux où la fonction objective est formée de deux termes.

Le premier terme représente le coût déterministe associé à la première période (le présent), alors que le second terme est la valeur de la fonction de recours qui représente le coût espéré s'étalant de la période suivante jusqu'à la fin de l'horizon (le futur).

L'auteur enchaîne par une formulation du problème comme un processus de décision markovien. Il s'agit alors d'identifier pour chaque véhicule une stratégie optimale qui vise la maximisation du profit moyen, en prenant l'une des décisions suivantes : transporter une charge de i à j, se déplacer sans charge de i vers j ou rester en place jusqu'à la prochaine période. La taille de l'espace des états ainsi que la complexité du calcul des probabilités de transition constituent une difficulté majeure du problème. Notons que dans le cas d'un seul véhicule, l'auteur fournit une stratégie optimale.

En 1996, l'auteur propose dans le même contexte un modèle hybride basé sur un réseau dynamique stochastique [56]. Ce réseau est formé de deux parties. La première est le réseau d'affectation chargé d'affecter les chauffeurs aux requêtes connues durant la période courante. La seconde partie correspond à une "projection" dans le futur et représente toutes les demandes connues mais non encore servies, ainsi que toutes les demandes prévues. Les arcs de cette partie du réseau représentent les déplacements des camions chargés ou vides ainsi que les temps d'attente entre deux périodes consécutives dans une même région. D'autre part, les déplacements des camions vides en vue d'anticiper les demandes futures sont représentés par un ensemble d'arcs entre les deux parties du réseau. La valeur espérée de la fonction de recours est modélisée par un regroupement de "liens de recours" qui représentent la contribution marginale espérée des véhicules disponibles dans une région donnée, durant une période donnée. Une approximation de

cette valeur est ajoutée aux coûts. Le problème de transport ainsi obtenu à chaque période est résolu par l'algorithme du simplexe spécialisé pour les problèmes de réseaux.

Récemment, Powell et al. ([58]) ont proposé une nouvelle approche de résolution pour des problèmes dynamiques stochastiques de grande taille. Les auteurs commencent par présenter l'approche dans un cadre général. Par la suite, ils rapportent son application dans différents contextes, comme la répartition des véhicules en temps réel. L'approche proposée incorpore des techniques de programmation dynamique et de programmation stochastique à niveaux multiples. Plus précisément, cette approche utilise les techniques de programmation dynamique vers l'avant [10] qui, contrairement aux techniques de programmation dynamique classiques, progressent vers l'avant pour éviter d'évaluer l'espace des états de façon extensive. Les auteurs contournent ainsi le problème de dimensionnement qui est inhérent aux approches de programmation dynamique.

2.6.3 Méthodes basées sur la théorie des files d'attente

Tout comme pour les problèmes avec requêtes à un point, peu de travaux adoptant ce genre d'approches sont rapportés dans la littérature.

Swihart et Papastavrou [87] introduisent une nouvelle variante du problème dynamique du réparateur, antérieurement introduit par Bertsimas et Van Ryzin [7, 8]. Dans cette variante, chaque requête de service est caractérisée par un point de collecte et un point de livraison. L'objectif est de minimiser l'espérance du temps que les requêtes passent dans le système. Les auteurs considèrent le cas de capacité unitaire (c-à-d., le véhicule ne peut transporter qu'une seule demande à la fois) ainsi que le cas où le véhicule peut transporter un nombre illimité de

requêtes simultanément. Dans chacun de ces deux cas, les auteurs utilisent des résultats de la théorie de files d'attente afin d'identifier des bornes inférieures pour le temps moyen passé dans le système. En outre, ils proposent et analysent différentes stratégies locales.

2.6.4 Autres approches

Le problème de répartition des véhicules en temps réel est caractérisé par la présence de contraintes nombreuses et très complexes. Toutefois, les planificateurs humains sont capables de prendre de bonnes décisions, sans doute grâce à l'expérience accumulée au fil des années. Dans le but d'imiter leur comportement, de nouvelles méthodes comme les réseaux de neurones et les algorithmes génétiques ont été utilisées.

Shen et Potvin [82] proposent un modèle de réseaux de neurones pour un problème de courrier rapide urbain. Le réseau de neurones suggère la route qui servira la nouvelle requête suite à une phase d'apprentissage à partir d'exemples de décisions provenant d'un planificateur professionnel. Notons que la difficulté principale dans le cas de modèles de réseaux de neurones consiste à ajuster adéquatement certains paramètres de l'algorithme qui sont au coeur de la phase d'apprentissage. En outre, le processus d'apprentissage nécessite une banque de données ainsi que les décisions correspondantes prises par des planificateurs expérimentés.

Benyahia et Potvin [6] appliquent la programmation génétique pour le même problème. Ils définissent un programme comme une fonction d'utilité qui sert à évaluer un véhicule en fonction d'un ensemble de critères (p.e. nombre de requêtes en retard suite à une nouvelle insertion, détour,...). Partant d'une population initiale de programmes, l'approche consiste à choisir les meilleurs et à les combiner

entre eux afin d'obtenir une nouvelle génération de programmes. Ce processus est répété pour un nombre fini de générations ou jusqu'à ce que le système ne s'améliore plus. La solution finale est le meilleur programme généré.
L'algorithme proposé donne de meilleurs résultats que ceux obtenus par Shen et Potvin [82]. Toutefois, l'approche souffre de la même faiblesse que les réseaux de neurones, à savoir la difficulté d'interprétation des programmes obtenus. En effet, la fonction d'utilité générée par l'algorithme est typiquement une fonction mathématique très complexe.

2.7 Conclusion

Nous avons présenté dans ce chapitre un survol des travaux reliés aux problèmes de répartition de véhicules en temps réel. En faisant abstraction de certains détails très pointus, nous avons tenté de regrouper ces travaux en un certain nombre de contextes d'application assez généraux. Nous avons établi par la suite une taxinomie permettant de caractériser chacun de ces contextes afin de pouvoir identifier des méthodologies appropriées pour chaque classe de problèmes. Finalement, nous avons discuté et comparé les principales approches de résolution rapportées dans la littérature. Nous en dégageons les conclusions suivantes :

- Lorsque le degré de dynamisme du problème est très faible, il peut être acceptable de considérer une adaptation d'algorithme statique même si le temps d'exécution de celui-ci est élevé. En effet, dans le cas où le degré de dynamisme est très faible, les données changent très rarement au cours du temps ce qui laisse plus de temps à l'exécution de l'algorithme statique.
- Les approches stochastiques sont très appropriées pour traiter l'aspect dynamique et stochastique des problèmes en temps réel. Cependant, ces méthodolo-

gies demeurent encore peu populaires et leurs progrès, quoiqu'indéniables, ne se font pas aussi rapidement que souhaités.

- Les approches hybrides combinant les techniques d'optimisation combinatoire et la théorie des files d'attente représentent une génération de méthodologies en émergence qui nous semblent très prometteuses. Ces approches sont particulièrement efficaces lorsque le degré de dynamisme est très élevé.
- Les pressions temporelles inhérentes à un environnement dynamique demeurent le principal obstacle aux approches "traditionnelles" basées sur l'adaptation d'algorithmes statiques. Les développements récemment accomplis dans le domaine de l'informatique, et en particulier dans le domaine du calcul parallèle, offrent un outil efficace pour faire face à cet obstacle. Cependant, cet outil est encore sous-utilisé.

Bibliographie

[1] **P. Badeau, M. Gendreau, F. Guertin, J.Y. Potvin, E. Taillard,** *A Parallel Tabu Search Heuristic for the Vehicule Routing Problem with Time Windows*, Transportation Research **5c** (1997), 109-122.

[2] **M.O.Ball, T.L. Magnanti, C.L. Monna and G.L. Nemhauser (eds),** *Network Routing*, Handbooks in Operations Research and Management Science, Volume 8, North-Holland : Amsterdam (1995).

[3] **D.O. Baush, G. G. Brown, D. Ronen,** *Consolidating and Dispatching Truck Shipment of Mobil Heavy Petroeum Products*, Interfaces **25** (1995), 01-17.

[4] **W.J. Bell, L.M. Dalberto, M.L. Fisher, A.J. Greenfield, R. Jaikumar, P. Kedia, R.G. Macj, P.J. Prutzman,** *Improving the Distribution of Industrial Gases with an On-Line Computrized Routing and Scheduling Optimizer* , Interfaces **13** (1983), 04-23.

[5] **R.E. Bellman,** *On a Routing Problem* , Quarterly of Applied Mathematics **16** (1958), 87-90.

[6] **I. Benyahia and J.Y. Potvin,** *Decision Support for Vehicule Dispatching Using Genetic Programming* , IEEE Transactions on Systems, Man and Cybernatics **28** (1998), 306-314.

[7] **D. Bertsimas and G. Van Ryzin,** *A Stochastic and Dynamic Vehicule Routing Problem in the Euclidian Plane*, Operations Research **39** (1991), 601-615.

[8] **D.J. Bertsimas and G. Van Ryzin,** *Stochastic and Dynamic Vehicule Routing in the Euclidian Plane with Multiple Capacitated vehicules*, Operations Research **41** (1993), 60-76.

[9] **D.J. Bertsimas and D. Simchi-Levi,** *A New Generation of Vehicle Routing Research : Robust Algorithms, Addressing Uncertainty*, Operations Research **44** (1996), 286-304.

[10] **D. Bertsekas and J. Tsitsiklis,** *"Neuro-Dynamic Programming"*, Athena Scientific, Belmont, MA. (1996).

[11] **J.R. Birge and J.K. Ho,** *Optimal Flows in Stochastic Dynamic Networks with congestion*, Operations Research, **41** (1993), 203-216.

[12] **J.Y. Blais, J.M. Rousseau,** *HASTUS : A Model for the Economic Evaluation of Drivers' Collective Agreements in Transit Companies*, INFOR **20** (1982), 03-15.

[13] **L. Bodin, B.L. Golden, A. Assad and M.O. Ball,** *Routing and Scheduling of Vehicules and Crews : the State of the Art*, Computers and Operations Research **10** (1983), 62-212.

[14] **J.W. Braklow, W.W. Graham, S.M. Hassler, K.E. Peck and W.B. Powell,** *Interactive Optimization Improves Service and Performance for Yellow Freight System*, Interfaces **22** (1992), 147-172.

[15] **L. Brotcorne, L. Farand, G. Laporte and F. Semet,** *Impacts des Nouvelles Technologies sur la Gestion des Systèmes de Véhicules d'Urgence*, Tech-

nical Report CRT-99-39, Centre de recherche sur les transports, Université de Montréal, Canada (1999).

[16] **G.G. Brown, C. J. Ellis, G. Lenn, W. Graves, David Ronen,** *Real Time, Wide Area Dispatch of Mobil Tank Trucks* , Interfaces **17** (1987), 107-120.

[17] **C.F. Daganzo,** *An Approximate Analytic Model of Many-To-Many Demand Responsive Transportation Systems* , Transportation Reaserch **12** (1978), 325-333.

[18] **M. Desrochers, J. Desrosiers and M. Solomon,** *A New Optimization Algorithm for the Vehicule Routing Problem with Time Windows*, Operations Reaserch **40** (1992), 342-354.

[19] **J. Desrosiers, Y. Dumas, M.M. Solomon and F. Soumis,** *Time Constrained Routing and Scheduling*, in Handbooks in Operations Research and Management Science, Volume 8 : Network Routing, M.O. Ball, T.L. Magnanti, C.L. Monma and G.L. Nemhauser (eds), pp. 35-139, North-Holland : Amsterdam (1995).

[20] **Robert B. Dial,** *Autonomious Dial-A-Ride Transit Intoductory Overview*, Transportation Reaserch **03C** (1995), 261-275.

[21] **M. Dror and W. Powell,** *Special Issue in Stochastic and Dynamic Models in Transportation* ,Operations Research **41** (1993), 11-17.

[22] **Y. Dumas,** *Confection d'Itinéraires pour le Transport Adapté*, Publication G-88-44, GERAD, H.E.C., Canada (1988).

[23] **M.L. Fisher, A. J. Greenfield, R. Jaikumar and P. Kedia,** *Real Time Scheduling of a Bulk-Delivery Fleet : Practical Application of Lagrangian Re-*

laxation, Report 82-10-11, Decision Sciences Dept., University of Pennsylvania (1982).

[24] **M.L. Fisher, H.J. Gang and T.B. Xing,** *Scheduling Bulk-Pickup-Delivery Vehicules in Shangai* , Interfaces **16** (1986), 18-23.

[25] **M. Gendreau, P. Badeau, F. Guertin, J.Y. Potvin, É. Taillard,** *A Solution Procedure for Real-time Routing and Dispatching of Commercial Vehicules*, Technical Report CRT-96-24, Centre de recherche sur les transports, Université de Montréal, Canada (1996).

[26] **M. Gendreau, A. Hertz, G. Laporte and M. Stan,** *A Generalized Insertion Heuristic for the Traveling Salesman Problem with Time Windows*, Operations Research **43** (1998), 330-335.

[27] **M. Gendreau, F. Guertin, J.Y. Potvin and R. Séguin,** *Neighborhood Serach Heuristics for a Dynamic Vehicle Dispatching Problem with Pick-Ups and Deliveries*, Technical Report CRT-98-10, Centre de recherche sur les transports, Université de Montréal, Canada (1998).

[28] **M. Gendreau, F. Guertin, J.Y. Potvin and É. Taillard,** *Parallel Tabu Search for Real-Time Vehicle Routing and Dispatching*, Transportation Science **33 (4)** (1999), 381-390.

[29] **M. Gendreau, G. Laporte and F. Semet,** *Solving an Ambulance Location Model by Tabu Search*, Location Science **5** (1997), 75-88.

[30] **M. Gendreau, G. Laporte and F. Semet,** *A Dynamic Model and Parallel Tabu Search Heuristic for Real-Time Ambulance Relocation*, Technical Report CRT-2000-17, Centre de recherche sur les transports, Université de Montréal. Canada (2000).

[31] **B.E. Gillett,** *A Heuristic Algorithm for the Vehicule-Dispatch Problem*, Operations Research **22** (1974), 340-349.

[32] **F. Glover,** *Tabu Search, PartI,*ORSA journal of computing **1** (1989), 190-206.

[33] **F. Glover,** *Tabu Search, PartII,*ORSA journal of computing **2** (1990), 04-32.

[34] **W.G. Graves, and B. A. Whinston,** *An Algorithm for the Quadratic Assignment Problem*, Management Science, **17** (1970), 453-471.

[35] **A. Hill, V. Mabert and D. Montgomory,** *A Decision Support System for the Courrier Vehicule Scheduling Problem*, Omega Int. J. of Mgmt Sci. **16** (1988), 333-345.

[36] **J.N. Hooker and N.R. Natraj,** *Solving a General Routing and Acheduling Problem by Chain Decomposition and Tabu Search*, Transp. Science **29** (1995), 30-44.

[37] **P. Jaillet,** *The Probabilistic Traveling Salesman Problem*, Ph.D. Thesis, Department of Civil Engineering, MIT (1985).

[38] **P. Jaillet,** *Probabilistic Routing Problems in the Plane*, Operational Research **29** (1991), 675-688.

[39] **I.Z. Jarrah and G. Yu,** *A Decision Support Framework for Airline Flight Cancelations and delays*, Transp. Science **27** (1993), 266-280.

[40] **J.J. Jaw, A.R. Odoni, H.N. Psaraftis and N.H.M. Wilson,** *A Heuristic Algorithm for the Multi-Vehicule Many-To-Many Advance Request Dial-A-Ride Problem*, Working Paper MIT-UMTA-82-3, M.I.T., Cambridge Massachusetts (1982).

[41] **J.J. Jaw, A.R. Odoni, H.N. Psaraftis and N.H.M. Wilson,** *A Heuristic Algorithm for the Multi-Vehicule Advance Request Dial-A-Ride Problem with Time Windows*, Transp. Res. **20B** (1986), 243-257.

[42] **B. Kalantani et al.,** *An Algorithm for the Traveling Salesman Problem with Pickup and Delevery Customers*, Journal of Operational Research **22** (1985), 377-386.

[43] **J.G. Klincewicz and B.M. Rosenwein,** *The Airline Exception Scheduling Problem*, Transportation Science **29** (1995), 04-16.

[44] **A. Larsen,** *The Dynamic Vehicle Routing Problem*, Ph.D. Dissertation, Technical University of Denmark, 2000.

[45] **K. Lund, O.B.G. Madsen, J.M. Rygaard,** *Vehicle Routing Problems with Varying Degrees of Dynamism* , Technical Report IMM-REP-1996-1, Institute of Mathematical Modeling, Technical University of Denmark, Lyngby, Denmark (1996).

[46] **O.B.G. Madsen, H.F. Racen and J.M. Rygaard,** *A Heuristic algorithm for a Dial-A-Ride Problem with Time Windows, Multiple Capacities and Multiple Objectives*, Annals of Op. Res. **60** (1995), 193-208.

[47] **O.B.G. Madsen, K. Tosti and J. Voelds,** *A Heuristic Method for Dispatching Repair Men*, Annals of Op. Res. **61** (1995), 213-226.

[48] **H.S. Mahmassani, Y. Kim and P. Jaillet,** *Local Optimization Approaches to Solve Dynamic Commercial Fleet Management Problems*, Transportation Research Record **1733** (2000), 71-79.

[49] **H. Min,** *The Multiple Vehicle Routing Problem with Simultaneous Delivery and Pickup points*, Transp. Res. **23A** (1989), 377-386.

[50] **A. Minkoff,** *A Real Time Dispatching of Delevery Vehicles*, Technical Report, Massachusetts Institute of Technology, Cambridge, Mass. (1985).

[51] **A.S. Minkoff,** *A Markov Decision Model and Decomposition Heuristic for Dynamic Vehicle Dispatching* , Operations Research, **41** (1993), 77-90.

[52] **J.Y. Potvin and J.M. Rousseau,** *Constraint-Direct Search for the Advanced Request Dial-A-Ride Problem with Service Quality Constraints*, in Computer Science and Operations research : New Developpement in the Interfaces, Balci, Sharda and Zenios (eds), pp. 457-474, Pergamon Press (1992).

[53] **W. Powell,** *A Comparative Review of Alternative Algorithms for the Dynamic Vehicle Allocation Problem*, in Vehicle Routing : Methods and Studies, B.L. Golden and A.A. Assad (eds), pp. 249-291, North Holland : Amsterdam (1988).

[54] **W. Powell, Y. Sheffi, K.S. Nickerson, K. Butterbaugh and S. Atherton,** *Maximizing Profits for North American Van Lines Truckload Division : A New Framework for Pricing and Operations*, Interfaces, **18** (1988), 21-41.

[55] **W.B. Powell, P. Jaillet and A. Odoni,** *Stochastic and Dynamic Networks and Routing*, in Network Routing, Handbooks in Operations Research and Management Science, Volume 8, M.O. Ball, T.L. Magnanti, C.L. Monma and G.L. Nemhauser (eds), pp. 141-295, North-Holland : Amsterdam (1995).

[56] **W.B. Powell,** *A Stochastic Formulation of the Dynamic Assignment Problem, with an Application to Truckload Motor Carriers*, Transportation Science **30 (3)** (1996), 195-219.

[57] **W. Powell,** *On Languages for Dynamic Resource Scheduling Problems*, in Fleet Management and Logistics, T.G. Crainic and G. Laporte (eds), pp. 127-157, Kluwer Academic Publishers (1998).

[58] **W.B. Powell, G. Godfrey, K. Papadaki, M. Spivey and H. Topaloglu,** *Adaptive Dynamic Programming for Multistage Stochastic Resource Allocation*, Department of Operations Research and Financial Engineering. Princeton University (2000).

[59] **W.B. Powell, W. Snow and R.K. Cheung,** *Adaptive Labeling Algorithms for the Dynamic Assignment Problem*, Transportation Science **34** (2000), 50-66.

[60] **W.B. Powell, M.T. Towns and A. Marar,** *On the value of Optimal Myopic Solutions for Dynamic Routing and Scheduling Problems in the Presence of User Non Compliance*, Transportation Science **34** (2000), 67-85.

[61] **H.N. Psaraftis,** *A Dynamic Programming Solution to the Single-Vehicule Many-to-Many Immediate Request Dial-A-Riade Problem* , Transportation Science **14** (1980), 130-154.

[62] **H.N. Psaraftis,** *K-Interchange Procedures for Local Search in a Precedence-Constrained Routing Problem*, European Journal of Operational Research **13** (1983), 391-402.

[63] **H.N. Psaraftis,** *An exact Algorithm for the Single Vehicule Many-To-Many Dial-A-Ride Problem with Time Windows*, Transportation Reaserch **17** (1983), 351-357.

[64] **H.N. Psaraftis,** *Analysis of $O(N^2)$ Heuristic for the Single Vehicule Many-To-Many Euclidian Dial-A-Ride Problem*, Trasportation Reaserch **17B** (1983), 133-145.

[65] **H.N. Psaraftis, J.B. Orlin, D. Bienstock and P.M. Thompson,** *Analysis and solution algorithms of sealift Routing and scheduling problems : Final Report*, Working Paper No. 1700-85, Sloan School of Management, M.I.T. (1985).

[66] **H.N. Psaraftis,** *Scheduling Large Scale Advance-Request Dial-A-Ride Systems*, Amer. J. Math. Management Sc. **06** (1986), 327-367.

[67] **H.N. Psaraftis,** *Dynamic Vehicule Routing Problems*, in Vehicule Routing : Methods and Studies, B.L. Golden and A.A. Assad (eds), pp. 223-249, North Holland : Amsterdam (1988).

[68] **H.N. Psaraftis M.M. Solomon, T.L. Magnanti and T.U. Kim,** *Routing and Scheduling on a Shoreline with release times*, Management Science **36** (1990), 212-223.

[69] **H.N. Psaraftis,** *Dynamic Vehicule Routing : Status and Prospects* , Annals of Operations Research **61** (1995), 143-164.

[70] **B. Ran, D.E. Boyce and L.J. Leblanc,** *A New Class of Indtantaneous Dynamic User-Optimal Traffic Assignmrnt Models*, Operations Research, **41** (1993), 192-202.

[71] **H. Rappoport,** *A Planing Heuristic for Military Airlift*, Interfaces **22** (1992), 73-87.

[72] **A.C. Regan, H.S. Mahmassani and P. Jaillet,** *Real-Time Information for Improved Efficiency of Commercial Vehicle Operations*, Proc., First World Congress on Applications of Transport Telematics & Intelligent Vehicle-Highway Systems, **Nov.** (1994).

[73] **A.C. Regan, H.S. Mahmassani and P. Jaillet,** *Improving Efficiency of Commercial Vehicle Operations Using Real-Time Information : Potential Uses and Assignment Strategies*, Transportation Research Record **1493**, 188-198 (1994).

[74] **A.C. Regan, H.S. Mahmassani and P. Jaillet,** *Dynamic Vehicle Allocation for Fleet Management : Operational Changes for Improved Efficiency*, Proc., 2nd World Congress on Applications of Transport Telematics & Intelligent Vehicle-Highway Systems, **Nov.** (1995).

[75] **A.C. Regan, H.S. Mahmassani and P. Jaillet,** *Dynamic Decision Making for Commercial Fleet Operations Using Real-Time Information*, Transportation Research Record **1537**, 91-97 (1995).

[76] **R. Rivard,** *Construction des parcours des véhicules et des Horaires des Chauffeurs pour le Transport des Personnes Handicapées*, publication CRT-240, Centre de recherche sur les transports, Université de Montréal, Canada (1981).

[77] **Y. Rochat and E. Taillard,** *Probabilistic Diversification and intensification in Local Search for Vehicule Routing*, Journal of Heuristics, **1** (1995), 147-167.

[78] **J.M. Rousseau, and S. Roy,** *RAO Répartition Assistée par Ordinateur : la Déscription du Prototype* , publication CRT-564, Centre de recherche sur les transports, Université de Montréal, Canada (1988).

[79] **S. Roy J.M. Rousseau, G. Lapalme and J.A. Ferland,** *Routing and Scheduling for the Transportation of Disables Persons : the Algorithm*, Publication CRT-412, Centre de recherche sur les transports, Université de Montréal, Canada (1985).

[80] **M.W.P. Savelsberg,** *The general Pickup and delevery Problem*, Tranp. Science **29** (1992), 17-29.

[81] **R. Séguin, J.Y. Potvin, M. Gendreau, T.C. Crainic and P. Marcotte** *Real-time Decision Problems : an Operational Research Perspective*, Journal of the Operational Research Society **48** (1997) 162-174.

[82] **Y. Shen, J.Y. Potvin, J.M. Rousseau and S. Roy,** *A Computer Assistant for Vehicule Dispatching with Learning Capabilities*, Annals of Operations Reaserch **61** (1995), 189-211.

[83] **M.M. Solomon,** *Algorithms for the Vehicule Routing and Scheduling Problems with Time Window Constraints*, Operations Research **35** (1987), 254-265.

[84] **R.S. Solanki,** *An execution planning Algorithm for Military Airlift*, Interfaces **21** (1991), 121-131.

[85] **D.M. Stein,** *Scheduling Dial-A-Ride Transportation Systems*, Transportation Reaserch **12** (1978), 232-249.

[86] **C. Sutcliffe and J. Board,** *Optimal Solution of a Vehicule Routing Problem : Transporting Mentally Handicapped Adults to an Adult Training Center*, J. Opl Res. Soc. **41** (1990), 61-67.

[87] **M.R. Swihart and J.D. Papastavrou,** *"A Stochastic and Dynamic Model for the Single-Vehicle Pick-up and Delivery Problem"*, Eur. J. Oper. Res. **114**, 447-464 (1999).

[88] **É. Taillard, P. Badeau, M. Gendreau, F. Guertin and J. Y. Potvin,** *A Tabu Search Heuristic for the Vehicle Routing Problem with Soft Time Windows*, Transportation Science **31** (1997) 170-186.

[89] **P.M. Thompson and H.N. Psaraftis,** *Cyclic Transfer Algorithms for Multivehicule Routing and Scheduling Problems*, Operations Research **41** (1993), 935-946.

[90] **P. Trudeau, J.M. Rousseau, J.A. Ferland and J. Choquette,** *An Operations Research Approch for the Planning and Operating of an Ambulance Service*, INFOR **27** (1989), 95-113.

[91] **J. Yang, P. Jaillet and H.S. Mahmassani,** *On-line Algorithms for Truck Fleet Assignment and Scheduling under Real-Time Information*, Transportation Research Record **1667** (1999), 107-113.

[92] **I. Webb,** *Period and Phase of Customer Replenishment : A New Approch to Inventory/Routing Problems*, Technical Report, Massachusetts Institute of Technology, Cambridge, Mass. (1989).

[93] **N.H.M. Wilson, J.M. Sussman, H.K. Wang and B.T. Higonnet,** *Scheduling Algorithms for Dial-a-Ride Systems*, Report USL TR-70-13, Urban Systems Laboratory. M.I.T. (1971).

[94] **N.H.M. Wilson and N.H. Colvin,** *Computer Control of the Rochester Dial-a-Ride System*, Report R77-31, Dept. of Civil Engineering. M.I.T. (1977).

Chapitre 3

Diversion Issues in Real-Time Vehicle Dispatching

La référence de cet article est :

S. Ichoua, M. Gendreau and J.Y. Potvin, "Diversion Issues in Real-Time Vehicle Dispatching", Transportation Science 34(4), 426-438 (2000).

Diversion Issues in Real-Time Vehicle Dispatching

Soumia Ichoua[1,2], Michel Gendreau[1,2]

and Jean-Yves Potvin[1,2]

1 Centre de recherche sur les transports. Université de Montréal, C.P. 6128, succ. Centre-ville, Montréal, Québec, Canada H3C 3J7.

2 Département d'informatique et de recherche opérationnelle, Université de Montréal, C.P. 6128, succ. Centre-ville, Montréal, Québec, Canada H3C 3J7.

Abstract. Recent technological advances in communication systems now allow the exploitation of real-time information for dynamic vehicle routing and scheduling. It is possible, in particular, to consider diverting a vehicle away from its current destination in response to a new customer request. In this paper, a strategy for assigning customer requests that includes diversion is proposed and various issues related to it are presented. An empirical evaluation of the proposed approach is performed within a previously reported tabu search heuristic. Simulations compare the tabu search heuristic, with and without the new strategy, on a dynamic problem motivated from a courier service application. The results demonstrate the potential savings that can be obtained through the application of the proposed approach. In particular, a reduction in the objective value that ranges from 0.8 % to 4.3 % is observed.

Introduction

In the past few years, there has been a rapid growth in communication and information technologies (e.g., global positioning satellites, cellular phones, geographic information systems, geosynchronous satellite-based systems, etc.). These recent advances afford opportunities for using real-time information to enhance the performance of decision systems in the area of vehicle routing. According to the fraction of requests that are known in advance, vehicle routing problems may be classified as static or dynamic. In the static case, all data are known before the routes are constructed and do not change afterwards (e.g., location of transportation requests, demand, etc...). However, in the dynamic case, all or a fraction of all requests are revealed as the routes are executed. Hence, dispatchers are forced to react to events that occur in real time, such as new service requests, unexpected delays, accidents, etc... Numerous examples of dynamic vehicle routing and dispatching problems may be found in practice, like ambulance or police services, courier services, dial-a-ride problems (e.g., transportation-on-demand for the handicapped) and many others.

Due to the new information technologies mentioned above, real-time information is now available at lower costs. This explains the new interest that dynamic vehicle routing problems have gained recently. However, there is still a lack of methodologies that can efficiently solve dynamic vehicle routing problems through a judicious integration of real-time information.

In dynamic vehicle routing problems, one potential use of real-time information is to divert a vehicle away from its current destination to serve a request that just occurred in the vicinity of its current position. In this work, we propose an approach

for the dynamic assignment of new requests that includes diversion and we examine different ways of implementing it. An empirical evaluation is performed within the tabu search heuristic reported in Gendreau et al. (96b).

The problem considered is motivated from a courier service application found in the local operations of long-distance shipping services, where the mail is collected at different customers' locations and brought back to a central office for further processing and shipping. This problem belongs to the class of pick up (or delivery) only problems where a set of requests must be transported to (or from) a central depot. The goal is to design a set of minimum cost routes, originating and ending at the central depot, that satisfy the transportation requests. In a dynamic context, each new request is inserted in real time in the current set of planned routes, where a planned route is the sequence of requests that have been assigned to a vehicle but have not been serviced yet.

The paper is organized as follows. Section 1 presents a brief literature review dedicated to dynamic vehicle routing problems. Then, section 2 describes the problem considered in this study. Section 3 briefly introduces the original tabu search heuristic reported in (Gendreau et al. (96b)) that we subsequently modified to explore the benefits of a new dynamic assignment strategy. Section 4 presents and discusses several issues related to it. Section 5 explains how the original tabu search heuristic was modified to include it. Section 6 reports computational results based on different ways of implementing the proposed strategy within the tabu search heuristic. Finally, section 7 summarizes our findings.

3.1 Literature review

The earliest papers in the literature on dynamic vehicle routing and dispatching were presented in the seventies and were either application-oriented (Wilson et al. (77)) or analytical (Daganzo (78) and Stein (78)). By the end of the eighties, dynamic vehicle routing gained an increasing attention. Two major factors explain this tendency : new developments in information technologies and the need for decision systems that could exploit this information to better represent the real world. Interesting survey articles on dynamic vehicle routing can be found in Psaraftis (88, 95), Powell (95) and Lund (96).

Because real-time vehicle routing problems are NP-hard and quick response times are required, exact algorithms are not yet capable of handling problems of realistic sizes (Psaraftis (80, 83, 85), Dial (95)). This justifies the use of heuristics in real-time environments. According to how they deal with the dynamic aspects of the problem, the problem solving approaches reported in the literature can be classified in three major categories. They are reported in the following subsections.

3.1.1 Adaptation of static algorithms

This approach is based on the notion of a rolling horizon. As time unfolds, static problems are solved repeatedly over events found within an horizon of length L that extends from the current time t to $t+L$. Different strategies emerge when the length L of the horizon is modified. If L is very small, a myopic near-term strategy is observed. In some cases this strategy can provide near-optimal solutions (Bell et al.(83), Dial(95), Psaraftis(85), Powell(88)). On the other hand, if L is very large, the problem considered is richer but, as it contains long-term events, the solution obtained is typically weaker unless a fast and powerful solution procedure is used

(Trudeau et al.(89), Gendreau et al.(96a), Gendreau et al. (96b)). Note also that longer-term events may be postponed, since they are likely to lead to useless calculations and updates.

Adaptation of static procedures to dynamic vehicle routing problems can be divided in two classes :

(a) A sophisticated static problem-solving procedure, which typically involves a re-optimization of the routes, is applied each time an input update occurs. Several researchers have used this approach like Bell et al.(83), Hill et al.(88), Brown et al.(87), Psaraftis(80, 83, 85), Powell et al.(88) and Dial(95). The drawback of this approach is the amount of computation time resulting from repeatedly executing the static algorithm. This disadvantage is more dramatic when new events occur frequently and when the execution of the static algorithm requires more time.

(b) Fast local operations (e.g., insertion) are used for reacting to any input revision (Trudeau et al.(89), Wilson(77), Rousseau and Roy(88), Solonki et al.(91) and Madsen et al.(95)). This approach is easy to implement and is appropriate for a dynamic environment where time pressure is important (e.g. Lund et al.(96)). However, it is myopic since solutions are produced through consecutive insertions (while a complete reordering of the routes may lead to better solutions). To overcome this weakness, some authors combine local operations with re-optimization procedures. This is often achieved by executing a set of successive insertions followed by a local search (e.g., exchange procedures like 2-opt, see Lin(65)). For different applications reported in the literature, see Rivard(81), Roy et al.(85), Gendreau et al.(96a) and Gendreau et al.(96b).

All studies mentioned above ignore the potential benefits of considering the stochastic aspects of the problems and trying to forecast the future. Stochastic me-

thods are aimed at overcoming this weakness.

3.1.2 Stochastic methods

Real-time dispatching problems have a stochastic nature (e.g., accidents, congestion, unexpected changes in meteorological conditions, etc...). Stochastic methods can be viewed as a "natural" way to judiciously address these issues. The goal is to react properly to an event to insure a good quality of service to the customers disturbed by these events, while minimizing their undesirable impact on the whole system. Two major classes of stochastic approaches are reported in the literature : stochastic programming and Markov decision processes.

(a) Markov decision processes :

Formulations based on this modeling approach were proposed by Powell (88), Bertsimas and Van Ryzin (91, 93) and Minkoff (93). Unfortunately, Markov decision processes are confronted with the following limitations that often prevent them from being applied to complex real-world problems :**(i)** the state space grows quickly with problem size; **(ii)** simplifying assumptions are often made to make the model more tractable.

(b) Stochastic programming :

The only work that we are aware of in this category is the one done by Powell in his comparative review of dynamic vehicle allocation problems (Powell et al. (88)). The author proposed a hybrid model that combines insights from Markov decision processes and classical network formulations.

3.1.3 Other methods

A new generation of approaches arises that try to replicate the dispatcher's decision making process. This is achieved by automating the decision procedure based on previous decisions taken by a skilled dispatcher. Within this framework, Shen and Potvin(95) used a neural network to elaborate an expert consulting system for a dispatcher working in a courier service company. In the same context, Benyahia and Potvin(95) proposed an approach based on genetic programming.

3.1.4 Diversion strategies

Apart from the study of Regan et al.(94, 95), all approaches that we are aware of fix the current destination of each vehicle. However, diverting a vehicle away from its current destination may be quite beneficial. Technically, diversion is now possible due to recent advances in information technologies. Regan et al. (94, 95) have proposed different diversion strategies in the context of a truck-load carrier. These studies will be discussed in section 4.

3.2 Problem definition

The problem considered is a dynamic pick up (or delivery) problem motivated from courier service applications. The static version of the problem can be stated as follows. Given a fixed size fleet of m identical vehicles and a set of requests to be served, the goal is to find a set of minimum cost routes that service these requests. More precisely :

- The vehicle routes must originate from and terminate at a fixed depot
- Each vehicle services one route and the service point of each request is visited

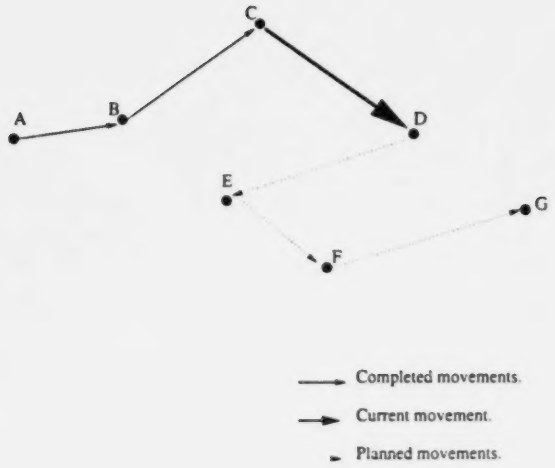

FIG. 3.1 – A vehicle route in a dynamic setting.

exactly once by exactly one vehicle ;

- No capacity constraint is considered (since small items are transported) ;
- Each service point, (including the depot) has its own time window $[e, l]$, where e is the earliest service time and l is the latest service time. Each route must start and end within the time window associated with the depot. Moreover, the service points have "soft" time windows. Thereby, a vehicle can arrive before the lower bound or after the upper bound : if the vehicle arrives too early, it must wait to start its service ; if the vehicle is too late, a penalty for lateness is incurred in the objective function.

The objective is to minimize a weighted sum of total distance and total lateness over all customers.

In the dynamic version of the problem, a number of service requests are not known completely ahead of time, but are rather dynamically revealed as time goes by. As illustrated in figure 3.1, in a dynamic setting, a vehicle route can be divided into three parts at any instant t :

- completed movements which form the part of the route already executed. Thereby, this part can not be modified anymore;
- current movement to reach the current destination;
- planned movements which constitute the portion of the routes not yet executed by the vehicle (planned route).

Given a new request at instant t, the problem is to assign this request to a particular vehicle and include it in its planned route at minimum cost.

In the following section, we briefly describe the original tabu search heuristic developed to tackle this problem.

3.3 A parallel tabu search algorithm

At any instant t, a solution to our dynamic problem is a set of planned routes, each beginning with the current destination of the associated vehicle. This current destination is fixed in the procedure to be described and can not be modified (i.e., the vehicle must reach its current destination).

The algorithm developed in Gendreau et al. (96b) is a parallel tabu search heuristic with an adaptive memory. Tabu search is an iterative local search technique that starts from some initial solution. At each iteration, a neighborhood is generated around the current solution and the best solution in this neighborhood becomes the new current solution (even if it does not provide an improvement). By allowing a degradation of the objective, it is possible to escape from bad local optima, as opposed to pure descent methods. The interested reader will find more details about this approach in Glover and Laguna (97).

The algorithm developed in Gendreau et al.(96b) can be described as follows :

- Construct I different initial solutions with a stochastic insertion heuristic

(where the choice of the next customer to be inserted is randomized).

- Apply tabu search to each solution and store the resulting routes in the adaptive memory.
- While a stopping criterion is not met do :
  - Use the routes stored in the adaptive memory to construct the initial solution.
  - Decompose the problem into subproblems obtained through a decomposition procedure.
  - Apply tabu search to each subproblem.
  - Add the routes of the resulting solution in the adaptive memory.

In the following, the main points of this algorithm are briefly introduced.

Adaptive memory

An adaptive memory which stores previously found elite solutions, is used to generate new starting points for the tabu search. This is achieved by combining routes taken from different solutions in memory (Rochat and Taillard (95)). Any new solution produced by the tabu search is included in the memory if it is not filled yet. Otherwise, the new solution replaces the worst solution in memory, if it is better.

Neighborhood structure of tabu search

The procedure for generating the neighborhood is called the CROSS exchange. Basically, two segments of variable lengths are taken from two different routes and are swapped.

Parallel implementation

To cope with real-time environments, the algorithm was implemented on a network of workstations. The parallelization of the procedure was achieved at two different

levels :

(1) Different tabu search threads run in parallel, each of them starting from a different initial solution.

(2) Within each search thread, many tabu searches run independently on subproblems obtained through a decomposition procedure.

Given the available platform, a master-slave scheme was chosen to implement the procedure. The master process manages the adaptive memory and creates initial solutions for the slave processes that run the tabu search.

A dynamic environment

Given the dynamic context, it is important to maintain the consistency of the adaptive memory with the current environment. Also, the memory itself must be consistent, that is, the current destination of each vehicle (which is fixed) must be the same for all solutions in memory. Thus, whenever a new event occurs, the search threads are first interrupted. Then, after an appropriate update of the adaptive memory, it is possible to restart the search threads with new solutions constructed from this updated memory. A new event may be of two types :

- A vehicle has finished serving its current customer. In this case, its next destination (i.e., the next customer to be serviced) has to be determined. The best solution in memory is used for this purpose. The remaining solutions in the adaptive memory are updated by removing this customer from its current location and by inserting it in first position in the planned route of the vehicle.
- A new service request has just occurred. In this case, the new request is inserted in all solutions in adaptive memory. If there is no feasible insertion position in any solution, the request is rejected. Otherwise, the request is

accepted and all solutions with no feasible insertion places are discarded from memory. The best solution is then re-optimized using a local search heuristic based on CROSS exchange in order to have at least one solution of high quality that includes the new request.

Section 4 will now consider the case where the next destination of each vehicle is not fixed. To address this issue, a dynamic assignment strategy that includes diversion as a special case is proposed. Its implementation within the tabu search heuristic presented here is reported in section 5.

3.4 A new strategy for handling customer requests

3.4.1 Motivation

Apart from the study of Regan et al.(94, 95), all approaches that we are aware of fix the current destination of each vehicle. However, diverting a vehicle away from its current destination may be quite beneficial. This could be considered, for example, when a new request occurs in the vicinity of the current position of some vehicle, while the driver is on his way to his current destination. Technically, diversion is now possible due to recent advances in communication technologies. In the following, we present an approach that includes diversion as a special case. We also investigate the trade-off between computation time and solution quality, given that this strategy takes place in a context where time pressure is important (vehicles are moving fast and diversion opportunities may be quickly lost).

In Regan et al.(94) and Regan et al. (95), the authors have proposed different diversion strategies in the context of a truck-load carrier. This problem is a combi-

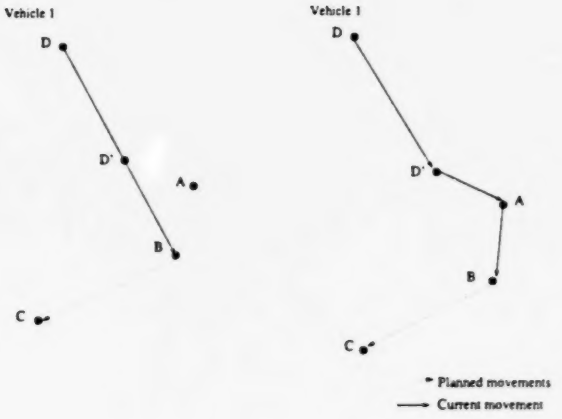

FIG. 3.2 – Diversion.

ned pick up and delivery problem with no consolidation (i.e. at any time, a vehicle is either empty or carrying a single load). In our application, on the other hand, a vehicle may be used to carry many loads concurrently. Also, in the context of a truck-load carrier, requests are more distant from each other and the activities take place over a longer time horizon (e.g., a few days). Finally, diversion was evaluated in a context where simple local dispatching rules were used, thus resulting in myopic approaches. The results reported in these studies show that substantial benefits can be obtained through the exploitation of diversion.

3.4.2 A broader view

Diversion consists of allowing a vehicle to be diverted away from its current destination to serve a request that just occurred in the vicinity of its current position. Figure 3.2 shows an example for one vehicle route. We suppose that a new request unfolds at point A at instant t, while vehicle 1 is at position D' on his way to service point B. The only modification that can result from en-route diversion is the modification of the current destination of one vehicle. This occurs

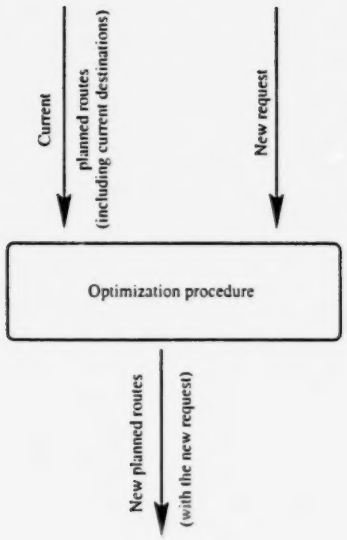

FIG. 3.3 – A dynamic assignment strategy that includes diversion.

if the vehicle services the new request before its planned current destination B. In this paper, we propose a broader approach where a change that occurs in the system may lead to redirecting one or more vehicles.

Figure 3.3 illustrates this new strategy in a schematic way. We first assume that the current destination of each vehicle is part of the planned route (and may thus change). Whenever a new request occurs, this request and the current routes are provided as input to some optimization procedure. The latter then produces a new set of routes with the new request. The optimization procedure, may be anything that we can think of, such as :

- A simple insertion procedure. Diversion then happen if the new request is inserted just before the current destination of some vehicle.
- An insertion followed by some re-optimization procedure (e.g., tabu search).
- An exact algorithm (assuming that it is fast enough).

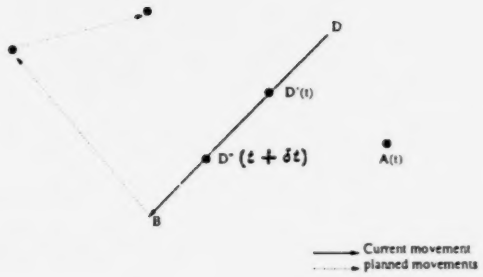

FIG. 3.4 – Time projection

Depending on the optimization procedure, it may well happen that the current destination of each vehicle remains the same after its application (although the planned routes have changed). Conversely, the current destination of one or more vehicles may change, but not necessary for servicing the new request (which appears further away on a planned route). Thus, diversion is simply a particular outcome of the dynamic assignment strategy presented here.

3.4.3 Solution quality versus computation time trade-off

Let us assume that a new request appears at point A at instant t and that it is possible to estimate the position D' of the vehicle (between the point D just served and the current destination B), see Fig. 3.4.

Since new opportunities for including the new request are now offered through the inclusion of the current destination in the planned route, the solution obtained should be better. However, some amount of time δt is required to apply the optimization procedure. Hence, a decision based on the situation at instant t when a new request appears, does not correspond to the state of the system at $(t + \delta t)$ when the decision becomes available (i.e. the vehicle will be at position $D"$ rather

than D'). Consequently, the situation should be assessed at instant $(t + \delta t)$ rather than instant t. The goal is to provide a decision that will fit in the context found at the end of the time allocated to the optimization procedure.

Clearly, this amount of time is a very important factor. In dynamic settings, finding a value that achieves a good trade-off between execution time and solution quality is very challenging. If δt is too small, not enough time is available for the optimization procedure. Therefore, the situation may be incorrectly assessed, resulting in a bad decision. On the other hand, if δt is too large, opportunities for diversion may be lost.

In order to assess the benefits that can be gained from the application of our dynamic assignment strategy, the parallel tabu search heuristic developed by Gendreau et al. (96b) was used as the optimization procedure. The next section shows how the original algorithm was modified to implement it.

3.5 Implementation

3.5.1 Inclusion of the current destination

In the original implementation of the tabu search heuristic (Gendreau et al. (96b)) reported in Section 3, the current destination of each vehicle is a fixed node which is not part of the planned route to be optimized. We thus included the current destination of each vehicle in the planned route and use instead the current position of the vehicle as the fixed node. With this modification, the re-optimization procedure based on CROSS exchanges can now move the current destination around (as any other service point on the planned route).

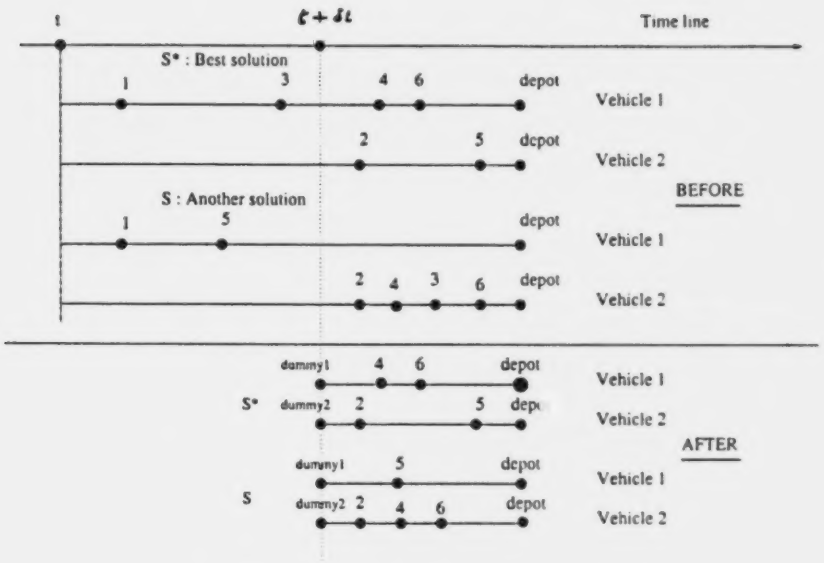

FIG. 3.5 – Update of solutions in the adaptive memory

3.5.2 Time projection

When a new request is received at time t, all solutions in the adaptive memory are updated according to the state of the system at time $(t + \delta t)$, where δt is the amount of time allocated to the optimization procedure. The planned routes of the best solution in adaptive memory are used for this purpose. A dummy point $F^i_{t+\delta t}$ is created for every vehicle route i of every solution in the adaptive memory. This dummy point $F^i_{t+\delta t}$ represents the position of vehicle i at time $(t + \delta t)$ and is the fixed node associated with route i.

Updating the solutions in the adaptive memory is illustrated in figure 3.5. In this figure, a fleet of two vehicles is considered : a new request occurs at instant t and δt is the amount of time allocated to the optimization procedure. In the best solution S^* at instant $t + \delta t$, vehicle 1 will have finished service at customer 1 and customer 3, while vehicle 2 will not have yet reached customer 2. Hence, before applying the

optimization procedure, customer 1 and customer 3 are removed from S^* and the dummy points are inserted at the beginning of the two routes. To update solution S, customer 1 and customer 3 are first removed from it (since they are serviced). Then the dummy fixed points are added at the beginning of the two routes. It is worth noting that the customers that have not been serviced yet are found in the same order on the planned routes before and after the update. Only the service times are likely to change.

3.5.3 Handling new events

Suppose that a new event occurs at instant t. The new events that occur during the time interval $[t, t+\delta t]$ will not interrupt the search threads. Instead, these events are handled as follows :

(a) When a vehicle has finished serving a customer, its next destination is determined by the best solution found in the adaptive memory. That is, in the time interval $[t, t + \delta t]$, everything is "frozen" and we simply follow the best solution in memory.

(b) When a new request occurs, we must decide whether to accept or reject it, since the customer may be waiting for a prompt answer about the ability to handle his request (a request is rejected if the end time of the planned route exceeds the upper bound of the time window at the depot after its insertion). Basically, the new request is inserted in all solutions in a copy of the adaptive memory. Solutions with no feasible insertion places are discarded. If at least one feasible solution remains, the new request is accepted and is kept in a list of pending requests. Otherwise, it is rejected.

At instant $(t+\delta t)$, all search threads are then interrupted. The adaptive memory is updated using the best solution found by the optimization procedure and the pending requests are handled one by one, as in the original algorithm (Gendreau et al.(96b)). Finally, the search threads are restarted with new solutions constructed from the adaptive memory.

3.5.4 Setting δt

The amount of time allocated to the optimization procedure must be related to the rate at which new events occur. When few events occur, a large evaluation time can be allowed. Conversely, in case of a highly dynamic environment, the evaluation must be done quickly. Based on these observations, three rules are considered.

Rule 1 : δt is chosen in such a way that the optimization procedure ends before any vehicle begins its service at its current destination. Let us assume that a service request is received at time t, and that service at the current destination on route i begins at time t_i. Then, choose δt such that : $t + \delta t \leq t_i$, $\forall i$, that is, $\delta t \leq min_i(t_i) - t$.

By choosing δt less than or equal to $min_i(t_i) - t$, we only need to add a dummy fixed node at the beginning of each planned route, which corresponds to the position of the vehicle at time $t + \delta t$. However, δt is often very small as there is typically one or more vehicles that are close to their current destination. Consequently, this rule was not retained in our computational experiments.

Rule 2 : The amount of time δt is fixed a priori and is related to the average inter-request time interval. This interval is proportional to a moving average of the l last inter-request intervals, that is $\delta t = \alpha_1 T^-$, where α_1 is a constant and T^- is the moving average. By relating δt to the average request

inter-arrival times, our choice is based on the intensity of occurrence of new requests.

Rule 3 : In rule 2, the computation of δt may take into account a service request that is not so urgent (i.e., its time window is much later than the current time). This is true, in particular, in contexts where the calls are received much before the requested service times. Therefore, another rule is proposed. Let X be the length of some time horizon and l_X the number of requests on the planned routes found within that horizon. Then the average time per request is defined as X/l_X and the interval δt is such that $\delta t = \alpha_2 X/l_X$, where α_2 is a constant.

Rule 2 and rule 3 represent two different strategies. In rule 2, intensity is measured using past events, since the moving average is calculated over the l last request inter-arrival times. In rule 3, intensity is assessed using events to come, since the average time per request is evaluated over requests found on the planned routes. Results obtained with these two strategies for setting δt are reported in the next section.

3.6 Computational results

3.6.1 Simulation framework

Simulations were performed to assess the benefits of the proposed approach. Data were taken from Solomon's 100-customer Euclidean problems (Solomon(87)), as in Gendreau et al. (96b). In the following, we describe the essential components of the simulator. The interested reader will find more details about this simulator in Gendreau et al.(96b).

- The customer locations are generated within a $([0, 100])^2$ square.
- The travel times are proportional to the corresponding Euclidean distances.
- Six different sets of problems are defined, namely C_1, C_2, R_1, R_2, RC_1, and RC_2. The customers are uniformly distributed in the problems of type R, clustered in groups in the problems of type C and mixed in the problems of type RC. In the problems of type 1, only a few customers can be serviced on each route due to a narrow time window at the depot, as opposed to problems of type 2 where each route may have many customers.
- The set of requests is divided into two subsets. The first subset contains requests that are assumed to be known at the start of the day. They are randomly selected among the entire set of requests with a bias in favor of requests with early time windows. The second subset contains requests that are received in real-time. In this case, the arrival time of each request is adjusted to create two different types of scenarios : scenario 1 with half of the requests known in advance and scenario 2 with only 25 % of the requests known in advance. In the latter case, a higher degree of dynamism is obtained, as there are more requests that occur in real-time within the same time horizon.
- Using minutes as time units for time-related data in Solomon's files, the time horizon is set to 15 minutes. The intent is to have a realistic rate of new unfolding requests (about 3 requests per minute for scenario 1 and 5 requests per minute for scenario 2).

3.6.2 Numerical results

The experiments reported in this section were performed on a network of 9 SUN UltraSparc-IIi workstations (300 MHz). Each process was programmed in C++ and communication between the processes was handled by the Parallel Virtual Machine Software. In these experiments, the objective function to minimize is the sum of the total distance traveled and the total tardiness over all customer locations. The fleet size was set to the number of routes in the best solution reported in the literature for each problem (Gendreau et al. (96b)). The value of the parameters found in the original algorithm were kept as in Gendreau et al. (96b). Parameters related to our new dynamic assignment strategy are tested in the following.

3.6.2.1 Preliminary tests

Scenario 1 (with half of the requests known in advance) was first considered for the preliminary experiments. The new algorithm was tested under rule 2 (see section 4) where the interval δt is computed as a multiple of a moving average T^- of inter-arrival times. That is, $\delta t = \alpha_1 T^-$, where α_1 is a multiplier. The moving average is taken over l loads, with $l = \beta_1 K$, K is the number of requests that are known in advance and β_1 is a multiplier. Parameters α_1 and β_1 are determined empirically.

Preliminary experiments were performed over 2 problems selected in each of the six categories C_1, C_2, R_1, R_2, RC_1, and RC_2. In these experiments, the values 0.10, 0.25, 0.50, 0.75 and 0.90 were tested for α_1 while the values 0.10 and 0.25 were tested for β_1. The results obtained did not clearly show an improvement over the original algorithm. Thus, more investigations were required, leading to an

important refinement in our algorithm.

As mentioned in section 5, a copy of the adaptive memory is used to decide whether a new request should be accepted or rejected. This can lead to inconsistencies with the true memory. In particular, two important problems were identified :

- A new best solution that is received from a tabu search thread is not inserted in the copy (only in the adaptive memory itself). Consequently, requests that could have been inserted in the solutions found in the adaptive memory at time $t + \delta t$ may have been wrongfully rejected.
- A new solution may be received from a tabu search thread, while the adaptive memory is full. In this case, the worst solution in adaptive memory is replaced by the new one. At this point, there is an inconsistency with the copy that still includes this worst solution.

To improve the implementation, both the adaptive memory and its copy are updated whenever a new solution is received from a tabu search thread. Table 1 shows the kind of improvement that can be obtained with the refined implementation, using $\alpha_1 = 0.9$ and $\beta_1 = 0.25$. The four numbers in each entry are the average number of unserved customers, distance traveled, total lateness and objective value (sum of total distance traveled and total lateness), respectively, for each problem set. The row "Overall" contains averages taken over the entire set of 56 test problems. Note that when the number of unserved customers is maintained, the objective value is improved. On the other hand, when the number of unserved customers is reduced, this is achieved with only a slight increase in the objective value.

3.6.2.2 Parameter values

To determine the best values for parameters α_1 and β_1 under rule 2, tests were performed over a small sample of problems selected in each of the six categories C_1, C_2, R_1, R_2, RC_1, and RC_2. Several values had to be tested for every problem in the sample. Thereby, a significant amount of computation time is required if the size of the sample is too large. On the other hand, this size has to be large enough to get a correct calibration. For these experiments, the size of the sample was set to 4. As scenario 1 and scenario 2 represent different degrees of dynamism (requests occur more frequently in scenario 2), the experiments were conducted separately for the two scenarios. Table 2 and table 3 report results obtained with different combinations of values using scenario 1 and scenario 2, respectively. As in Table 1, the four numbers in each entry are the average number of unserved customers, distance traveled, total lateness and objective value, respectively, for each problem set. Based on the results obtained, the values $\alpha_1 = 0.50$, $\beta_1 = 0.15$ for scenario 1 and $\alpha_1 = 0.50$, $\beta_1 = 0.10$ for scenario 2 were finally selected.

Under rule 3, the interval δt is computed as a function of the length of some rolling time horizon, that is, $\delta t = \alpha_2 X / l_X$, where α_2 is a constant, l_X is the number of requests on the planned routes found within the horizon X; $X = \beta_2 L$, where L is the length of the time horizon for the simulation and β_2 is a multiplier. The tests were conducted as for rule 2. Table 4 and table 5 report results obtained with different combinations of values using scenario 1 and scenario 2, respectively. The values $\alpha_2 = 0.50$, $\beta_2 = 0.25$ for scenario 1 and $\alpha_2 = 0.25$, $\beta_2 = 0.25$ for scenario 2 were finally selected.

Note that under rule 2, δt is about 5 seconds for scenario 1, and 2.8 seconds for scenario 2. Under rule 3, δt is about 4 seconds for scenario 1 and 2 seconds for

scenario 2.

3.6.2.3 Assessing the new implementation

Tables 6 and 7 compare the new algorithm (with the best parameter values found under rule 2 and rule 3) with the original algorithm reported in Gendreau et al.(96b), for scenario 1 and scenario 2, respectively. Here, the tests were conducted over all problems in each category. The results highlight the fact that the new implementation performs well against the original algorithm for the two scenarios. More precisely :

Rule 2 : For scenario 1, both the number of unserved customers and objective value were improved with regard to the original algorithm (apart from subset R_2). The reduction ranged from 33.3% to 100% for the number of unserved customers and from 0.6% to 3.0% for the objective value. In the case of subset R_2, the increase in the objective value was marginal when compared to the reduction in the number of unserved customers : 0.3% against 60%. Note that the lowest reduction observed in the objective value with the new strategy is still larger than the only increase observed (0.7% against 0.3%).

For scenario 2, the results demonstrate that applying our approach under rule 2 reduces both the number of unserved customers and the objective value in all categories. The reduction ranges from 11.2% to 67% for the number of unserved customers, and from 1.0% to 2.7% for the objective value.

Rule 3 : For scenario 1 and for categories C_1 and RC_2, where the original algorithm succeeds in serving all customers, the new algorithm under rule 3 improves the objective value. This reduction ranges from 1.3% to 4.3%. In the case of R_1 and C_2, both the number of unserved customers and objective value were improved.

The reduction ranged from 66.8% to 100% for the number of unserved customers, and from 2.6% to 2.7% for the objective value. For RC_1 and R_2, the number of unserved customers was improved under rule 3 at the cost of a small increase in the objective value (i.e., 67% against 0.04% for RC_1 and 80% against 0.6% for R_2). Using rule 3 instead of rule 2 results in a small reduction of the objective value ranging from 0.5% to 2.0% when the number of unserved customers is the same. Rule 3 succeeds in reducing the number of unserved customers, as obtained by rule 2 for RC_1 and R_2, at the cost of a small increase in the objective value (i.e. 50% against 1.8%, and 51% against 0.2%).

For scenario 2, the comparison with the original algorithm indicates a reduction of both the number of unserved customers and the objective value over all categories. This reduction ranges from 16.8% to 100% for the number of unserved customers, and from 2.0% to 4.3% for the objective value. The results also indicate that, apart from RC_1 and R_2, rule 3 leads to the same number of unserved customers than rule 2, but it reduces the objective value by 0.9% to 2.2%. For R_2 and RC_1, when compared with rule 2, rule 3 succeeds in reducing both the number of unserved customers (i.e. 12% and 100% respectively), and the objective value (i.e. 0.6% and 1.1% respectively).

3.7 Conclusion

In this paper, a new strategy for the dynamic assignment of new requests that includes diversion as a special case was proposed. Tests were conducted to assess its benefits within a previously reported tabu search heuristic. Some issues related to the proposed model were addressed and explored. Results show that a reduction in the sum of total distance traveled and total lateness, as well as in the number of

unserved customers is observed when compared with the original heuristic, where the current destination of each vehicle is fixed.

Future work will be aimed at considering additional sources of uncertainty like congestion, accidents, vehicle breakdowns, and others. A taxonomy of such events will be developed to help us in the task of determining appropriate responses.

Problem set	$\alpha_1 = 0.90, \beta_1 = 0.25$ *new*	$\alpha_1 = 0.90, \beta_1 = 0.25$ *old*
C_1	0 [a]	0
9	835.781 [b]	838.784
problems	2.025 [c]	1.403
	837.807 [d]	840.187
R_1	0.583	0.75
12	1183.00	1182.49
problems	51.125	52.55
	1234.12	1235.04
RC_1	0.375	0.25
8	1317.37	1316.12
problems	47.058	47.852
	1364.43	1363.97
C_2	0	0.25
8	608.321	602.624
problems	0	0
	608.321	602.624
R_2	0.181	0.181
11	1058.76	1073.06
problems	57.20	66.64
	1115.97	1139.70
RC_2	0	0
8	1188.09	1244.00
problems	45.166	53.276
	1233.26	1297.27
Overall	0.21	0.27
56	1040.62	1050.80
problems	35.69	39.02
	1076.31	1089.82

[a] Nb. of unserved customers.
[b] Total distance traveled.
[c] Total lateness.
[d] Objective value.

TAB. 3.1 – Comparison of the refined implementation (new) with the old one using scenario 1.

Problem set	$\alpha_1 = 0.90$ $\beta_1 = 0.15$	$\alpha_1 = 0.90$ $\beta_1 = 0.25$	$\alpha_1 = 1.20$ $\beta_1 = 0.25$	$\alpha_1 = 0.50$ $\beta_1 = 0.10$	$\alpha_1 = 0.50$ $\beta_1 = 0.15$
C_1	0 [a]	0	0	0	0
4	829.89 [b]	832.35	828.535	852.31	828.94
problems	0 [c]	0	0	0.65	0
	829.89 [d]	832.35	828.535	852.96	828.94
R_1	0.25	0	0.25	0.25	0
4	1253.02	1284.25	1262.32	1266.03	1281.79
problems	63.435	38.307	43.90	47.12	45.385
	1316.46	1322.56	1306.22	1313.15	1327.18
RC_1	0.5	0.75	0.5	0.75	0.25
4	1334.86	1324.50	1310.76	1318.16	1330.19
problems	74.472	62.985	55.962	67.762	83.022
	1409.33	1387.49	1366.72	1385.92	1413.22
C_2	0	0	0	0	0
4	642.978	625.628	612.602	616.238	627.51
problems	0	0	0	0	0
	642.978	625.628	612.602	616.238	627.51
R_2	0.5	0.25	0	0.75	0
4	1096.48	1106.52	1106.13	1126.82	1107.19
problems	24.182	30.155	19.337	53.782	14.645
	1120.66	1136.67	1125.47	1180.60	1121.84
RC_2	0	0	0	0	0
4	1226.10	1184.14	1188.22	1198.09	1180.53
problems	24.045	38.74	10.745	45.41	17.482
	1250.14	1222.88	1198.96	1243.50	1198.01

[a]Nb. of unserved customers.

[b]Total distance traveled.

[c]Total lateness.

[d]Objective value.

TAB. 3.2 – Searching for the best parameter values under rule 2 for scenario 1

Problem set	$\alpha_1 = 0.50$ $\beta_1 = 0.10$	$\alpha_1 = 0.50$ $\beta_1 = 0.15$	$\alpha_1 = 1.20$ $\beta_1 = 0.15$	$\alpha_1 = 0.90$ $\beta_1 = 0.25$	$\alpha_1 = 0.25$ $\beta_1 = 0.25$
C_1	0 [a]	0	0	0	0
4	882.42 [b]	871.055	908.478	925.06	881.658
problems	0.105 [c]	0	19.79	15.027	2.127
	882.525 [d]	871.055	928.268	940.088	883.785
R_1	0.25	0	0.25	0	0.50
4	1296.70	1284.24	1303.97	1303.56	1296.28
problems	76.00	63.982	73.315	67.71	53.067
	1372.70	1348.22	1377.28	1371.28	1349.35
RC_1	0.25	0.75	0.5	1.00	1.00
4	1366.52	1349.24	1405.42	1402.88	1354.10
problems	54.99	80.737	77.917	61.32	64.82
	1421.50	1429.98	1483.34	1464.20	1418.92
C_2	0	0.25	0	0	0
4	607.572	619.515	613.62	621.495	612.33
problems	0	0	0.002	0	0
	607.572	619.515	613.622	621.495	612.33
R_2	0	0.75	0	0.25	0
4	1113.66	1093.96	1119.40	1140.72	1106.98
problems	17.63	93.605	23.73	14.032	22.36
	1131.29	1187.56	1143.13	1154.76	1129.34
RC_2	0	0	0	0	0
4	1207.95	1235.45	1213.50	1144.22	1218.88
problems	37.50	19.827	35.867	60.822	37.622
	1245.45	1255.28	1249.37	1205.04	1256.50

[a]Nb. of unserved customers.

[b]Total distance traveled.

[c]Total lateness.

[d]Objective value.

TAB. 3.3 – Searching for the best parameter values under rule 2 for scenario 2

Problem set	$\alpha_2 = 0.50$ $\beta_2 = 0.15$	$\alpha_2 = 0.90$ $\beta_2 = 0.15$	$\alpha_2 = 1.20$ $\beta_2 = 0.15$	$\alpha_2 = 0.50$ $\beta_2 = 0.25$	$\alpha_2 = 0.25$ $\beta_2 = 0.25$
C_1	0 [a]	0	0	0	0
4	829.26 [b]	834.125	857.005	829.112	839.488
problems	0 [c]	0	0.375	0	0
	829.26 [d]	834.125	857.38	829.112	839.488
R_1	0.25	0.25	0.25	0.25	0.25
4	1272.71	1274.41	1267.60	1275.08	1265.19
problems	51.997	52.532	43.627	39.537	45.217
	1324.71	1326.94	1311.23	1314.62	1310.41
RC_1	0.75	0.75	1	0.75	0.75
4	1339.43	1344.64	1321.06	1325.66	1352.71
problems	62.587	74.707	71.185	60.04	64.92
	1402.02	1419.34	1392.25	1385.70	1417.63
C_2	0	0	0.5	0	0
4	615.518	614.925	616.928	616.878	624.21
problems	0	0	0	0	0
	615.518	614.925	616.928	616.878	624.21
R_2	0	0	0.5	0	0
4	1110.10	1095.61	1113.52	1091.81	1105.60
problems	60.165	61.09	46.065	56.442	64.455
	1170.27	1156.70	1159.58	1148.25	1170.06
RC_2	0	0	0	0	0
4	1186.05	1198.42	1172.54	1177.27	1198.94
problems	23.472	29.502	23.862	25.457	32.652
	1209.52	1227.92	1196.40	1202.73	1231.60

[a]Nb. of unserved customers.

[b]Total distance traveled.

[c]Total lateness.

[d]Objective value.

TAB. 3.4 – Searching for the best parameter values under rule 3 for scenario 1

Problem set	$\alpha_2 = 0.25$ $\beta_2 = 0.25$	$\alpha_2 = 0.50$ $\beta_2 = 0.25$	$\alpha_2 = 0.90$ $\beta_2 = 0.15$	$\alpha_2 = 1.20$ $\beta_2 = 0.15$	$\alpha_2 = 0.50$ $\beta_2 = 0.15$
C_1	0 [a]	0	0	0	0
4	880.675 [b]	889.275	880.555	955.958	882.895
problems	1.977 [c]	0	0.90	1.695	0
	882.652 [d]	889.275	881.455	957.652	882.895
R_1	0	0.25	0.25	0.25	0.25
4	1282.77	1302.15	1293.42	1282.50	1285.20
problems	46.415	51.71	60.865	58.432	56.23
	1329.18	1353.86	1354.29	1340.93	1341.42
RC_1	0	1.00	1.00	1.00	0.25
4	1366.24	1358.48	1380.77	1363.36	1334.95
problems	71.422	89.21	60.082	71.322	68.03
	1437.66	1447.69	1440.86	1434.68	1402.98
C_2	0	0.25	0	0	0
4	601.972	650.912	612.018	630.335	603.122
problems	0	0	0	0	0
	601.972	650.912	612.018	630.335	603.122
R_2	0	0.50	0	0.50	0.50
4	1089.10	1090.41	1083.09	1104.65	1101.28
problems	14.995	36.237	22.917	14.192	40.322
	1104.09	1126.65	1106.00	1118.84	1141.60
RC_2	0	0	0	0	0
4	1221.02	1207.80	1191.57	1207.86	1218.99
problems	35.692	28.942	40.562	46.522	27.662
	1256.72	1236.75	1232.13	1254.38	1246.66

[a]Nb. of unserved customers.

[b]Total distance traveled.

[c]Total lateness.

[d]Objective value.

TAB. 3.5 – Searching for the best parameter values under rule 3 for scenario 2

	Original algorithm	*New algorithm*	
Problem set		*rule2* $\alpha_1 = 0.50$ $\beta_1 = 0.15$	*rule3* $\alpha_2 = 0.50$ $\beta_2 = 0.25$
C_1	0 [a]	0	0
9	835.098 [b]	830.024	829.628
problems	5.535 [c]	3.341	0
	840.633 [d]	833.366	829.628
R_1	0.5	0.166	0.166
12	1219.42	1194.75	1178.54
problems	43.267	48.625	51.460
	1262.69	1243.38	1230.00
RC_1	0.375	0.25	0.125
8	1349.65	1304.58	1333.41
problems	44.22	64.955	60.965
	1393.87	1369.53	1394.38
C_2	0.125	0	0
8	612.798	609.232	597.011
problems	0.326	0	0
	613.124	609.232	597.011
R_2	0.454	0.181	0.09
11	1028.34	1035.09	1020.55
problems	31.96	29.053	46.334
	1060.30	1064.15	1066.88
RC_2	0	0	0
8	1203.75	1175.87	1160.83
problems	30.811	22.108	20.672
	1234.56	1197.98	1181.50
Overall	0.27	0.10	0.07
56	1049.83	1034.11	1027.95
problems	27.20	29.10	31.79
	1077.03	1063.22	1059.74

[a]Nb. of unserved customers.

[b]Total distance traveled.

[c]Total lateness.

[d]Objective value.

TAB. 3.6 – Comparison with the original algorithm for scenario 1

	Original algorithm	*New algorithm*	
Problem set		*rule2* $\alpha_1 = 0.50$ $\beta_1 = 0.10$	*rule3* $\alpha_2 = 0.25$ $\beta_2 = 0.25$
C_1	0 [a]	0	0
9	877.332 [b]	843.924	833.061
problems	10.455 [c]	11.227	8.027
	878.378 [d]	855.152	841.089
R_1	0.4	0.333	0.333
12	1226.21	1203.99	1197.30
problems	55.431	67.167	49.121
	1281.64	1271.16	1246.42
RC_1	0.375	0.125	0
8	1400.50	1358.78	1346.98
problems	56.391	73.28	69.002
	1456.89	1432.06	1415.98
C_2	0	0	0
8	610.775	600.796	595.635
problems	0	0	0
	610.775	600.796	595.635
R_2	0.818	0.727	0.636
11	1043.96	1043 43	1027.82
problems	216.696	193.885	207.639
	1260.65	1237.32	1235.46
RC_2	0	0	0
8	1222.22	1189.48	1184.77
problems	27.441	41.023	19.568
	1249.66	1230.51	1204.34
Overall	0.30	0.23	0.19
56	1070.75	1048.45	1039.11
problems	68.10	70.61	65.25
	1138.85	1119.06	1104.36

[a]Nb. of unserved customers.

[b]Total distance traveled.

[c]Total lateness.

[d]Objective value.

TAB. 3.7 – Comparison with the original algorithm for scenario 2

Bibliographie

[1] **W.J. Bell, L.M. Dalberto, M.L. Fisher, A.J. Greenfield, R. Jaikumar, P. Kedia, R.G. Macj, P.J. Prutzman,** *"Improving the Distribution of Industrial Gases with an On-Line Computrized Routing and Scheduling Optimizer"*, Interfaces **13**, 04-23 (1983).

[2] **I. Benyahia and J.Y. Potvin,** *"Decision Support for Vehicle Dispatching Using Genetic Programming"*, publication CRT-95-23, Centre de recherche sur les transports, Université de Montréal, Canada, 1995.

[3] **D. Bertsimas and G. Van Ryzin,** *"A Stochastic and Dynamic Vehicle Routing Problem in the Euclidian Plane"*, Operations Research **39**, 601-615 (1991).

[4] **D.J. Bertsimas and G. Van Ryzin,** *"Stochastic and Dynamic Vehicle Routing in the Euclidian Plane with Multiple Capacitated vehicles"*, Operations Research **41**, 60-76 (1993).

[5] **G.G. Brown, C. J. Ellis, G. Lenn, W. Graves, David Ronen,** *"Real Time, Wide Area Dispatch of Mobil Tank Trucks"*, Interfaces **17**, 107-120 (1987).

[6] **C.F. Daganzo,** *"An Approximate Analytic Model of Many-To-Many Demand Responsive Transportation Systems"*, Transportation Reaserch **12**, 325-

333 (1978).

[7] **Robert B. Dial,** *"Autonomous Dial-A-Ride Transit Intoductory Overview"*, Transportation Reaserch **03C**, 261-275 (1995).

[8] **M. Gendreau, P. Badeau, F. Guertin, J.Y. Potvin, É. Taillard,** *"A Solution Procedure for Real-time Routing and Dispatching of Commercial Vehicles"*, Technical Report CRT-96-24, Centre de recherche sur les transports, Université de Montréal, Canada, 1996a.

[9] **M. Gendreau, F. Guertin, J.Y. Potvin and É. Taillard,** *"Tabu Search for Real-Time Vehicle Routing and Dispatching"*, Technical Report CRT-96-47, Centre de recherche sur les transports, Université de Montréal, Canada, 1996b.

[10] **F. Glover and M. Laguna,** *Tabu Search*, Kluwer, 1997.

[11] **A. Hill, V. Mabert and D. Montgomory,** *"A Decision Support System for the Courrier Vehicle Scheduling Problem"*, Omega Int. J. of Mgmt Sci. **16**, 333-345 (1988).

[12] **S. Lin,** *"Computer Solutions of the Traveling Salesman Problem"*, Bell System Technical Journal **44**, 2245-2269 (1965).

[13] **K. Lund, O.B.G. Madsen, J.M. Rygaard,** *"Vehicle Routing Problems with Varying Degrees of Dynamism"*, Technical Report IMM-REP-1996-1, Institute of Mathematical Modeling, Technical University of Denmark, Lyngby, Denmark, 1996.

[14] **O.B.G. Madsen, H.F. Ravn and J.M. Rygaard,** *"A Heuristic algorithm for a Dial-A-Ride Problem with Time Windows, Multiple Capacities and Multiple Objectives"*, Annals of Op. Res. **60**, 193-208 (1995).

[15] **A.S. Minkoff,** *"A Markov Decision Model and Decomposition Heuristic for Dynamic Vehicle Dispatching"*, Operations Research, **41**, 77-90 (1993).

[16] **W. Powell,** *A Comparative Review of Alternative Algorithms for the Dynamic Vehicle Allocation Problem*, in Vehicle Routing : Methods and Studies, B.L. Golden and A.A. Assad (eds), pp. 249-291, North Holland : Amsterdam, 1988.

[17] **W. Powell, Y. Sheffi, K.S. Nickerson, K. Butterbaugh and S. Atherton,** *"Maximizing Profits for North American Van Lines Truckload Division : A New Framework for Pricing and Operations"*, Interfaces, **18**, 21-41 (1988).

[18] **W.B. Powell, P. Jaillet and A. Odoni,** *Stochastic and Dynamic Networks and Routing*, in Network Routing, Handbooks in Operations Research and Management Science, Volume 8, M.O. Ball, T.L. Magnanti, C.L. Monma and G.L. Nemhauser (eds), pp. 141-295, North-Holland : Amsterdam, 1995.

[19] **H.N. Psaraftis,** *"A Dynamic Programming Solution to the Single-Vehicle Many-to-Many Immediate Request Dial-A-Riade Problem"*, Transportation Science **14**, 130-154 (1980).

[20] **H.N. Psaraftis,** *"An exact Algorithm for the Single Vehicle Many-To-Many Dial-A-Ride Problem with Time Windows"*, Transportation Reaserch **17**, 351-357 (1983).

[21] **H.N. Psaraftis, J.B. Orlin, D. Bienstock and P.M. Thompson,** *"Analysis and solution algorithms of sealift routing and scheduling problems : Final Report"*, Working Paper No. 1700-85, Sloan School of Management, M.I.T., 1985.

[22] **H.N. Psaraftis,** *Dynamic Vehicle Routing Problems*, in Vehicle Routing : Methods and Studies, B.L. Golden and A.A. Assad (eds), pp. 223-249, North

Holland : Amsterdam, 1988.

[23] **H.N. Psaraftis,** *"Dynamic Vehicle Routing : Status and Prospects"*, Annals of Operations Research **61**, 143-164 (1995).

[24] **A.C. Regan, H.S. Mahmassani and P. Jaillet,** *"Real-Time Information for Improved Efficiency of Commercial Vehicle Operations"*, Proc., First World Congress on Applications of Transport Telematics & Intelligent Vehicle-Highway Systems, **Nov.** (1994).

[25] **A.C. Regan, H.S. Mahmassani and P. Jaillet,** *"Improving Efficiency of Commercial Vehicle Operations Using Real-Time Information : Potential Uses and Assignment Strategies"*, Transportation Research Record **1493**, 188-198 (1994).

[26] **A.C. Regan, H.S. Mahmassani and P. Jaillet,** *"Dynamic Vehicle Allocation for Fleet Management : Operational Changes for Improved Efficiency"*, Proc., 2nd World Congress on Applications of Transport Telematics & Intelligent Vehicle-Highway Systems, **Nov.** (1995).

[27] **A.C. Regan, H.S. Mahmassani and P. Jaillet,** *"Dynamic Decision Making for Commercial Fleet Operations Using Real-Time Information"*, Transportation Research Record **1537**, 91-97 (1995).

[28] **R. Rivard,** *"Construction des parcours des véhicules et des Horaires des Chauffeurs pour le Transport des Personnes Handicapées"*, publication CRT-240, Centre de recherche sur les transports, Université de Montréal, Canada. 1981.

[29] **Y. Rochat and E. Taillard,** *"Probabilistic Diversification and intensification in Local Search for Vehicle Routing"*, Journal of Heuristics, **1**, 147-167 (1995).

[30] **J.M. Rousseau, and S. Roy,** *"RAO Répartition Assistée par Ordinateur : la Déscription du Prototype"*, publication CRT-564, Centre de recherche sur les transports, Université de Montréal, Canada, 1988.

[31] **S. Roy J.M. Rousseau, G. Lapalme and J.A. Ferland,** *"Routing and Scheduling for the Transportation of Disables Persons : the Algorithm"*, Publication CRT-412, Centre de recherche sur les transports, Université de Montréal, Canada, 1985.

[32] **Y. Shen and J.Y. Potvin,** *"A Computer Assistant for Vehicle Dispatching with Learning Capabilities"*, Operations Reaserch **61**, 189-211 (1995).

[33] **R.S. Solanki,** *"An execution Planning Algorithm for Military Airlift"*, Interfaces **21**, 121-131 (1991).

[34] **D.M. Stein,** *"Scheduling Dial-A-Ride Transportation Systems"*, Transportation Reaserch **12**, 232-249 (1978).

[35] **P. Trudeau, J.M. Rousseau, J.A. Ferland and J. Choquette,** *"An Operations Research Approch for the Planning and Operating of an Ambulance Service"*, INFOR **27**, 95-113 (1989).

[36] **N.H.M. Wilson and N.H. Colvin,** *"Computer Control of the Rochester Dial-a-Ride System"*, Report R77-31, Dept. of Civil Engineering. M.I.T., 1977.

Chapitre 4

Vehicle Dispatching With Time-Dependent Travel Times

La référence de cet article est :

S. Ichoua, M. Gendreau and J.Y. Potvin, "Vehicle Dispatching with Time-Dependent Travel Times", soumis à EJOR en Février 2001.

Vehicle Dispatching With Time-Dependent Travel Times

Soumia Ichoua[1,2], Michel Gendreau[1,2]

and Jean-Yves Potvin[1,2]

1 Centre de recherche sur les transports. Université de Montréal, C.P. 6128, succ. Centre-ville, Montréal, Québec, Canada H3C 3J7.

2 Département d'informatique et de recherche opérationnelle, Université de Montréal, C.P. 6128, succ. Centre-ville, Montréal, Québec, Canada H3C 3J7.

Abstract. Most of the models for vehicle routing reported in the literature assume constant travel times. Clearly, ignoring the fact that the travel time between two locations does not depend only on the distance traveled, but on many other factors including the time of the day, impact the applicability of these models to real-world problems. In this paper, we present a model based on time-dependent travel speeds. An experimental evaluation of the proposed model is performed in a static and a dynamic setting, using a parallel tabu search heuristic. The results demonstrate that the proposed model provides substantial improvements over a model based on fixed travel times.

Keywords. Tranportation, Vehicle routing, Time-dependent travel speeds, Static and dynamic setting, Tabu search, Parallelism.

4.1 Introduction

The routing and scheduling of a fleet of vehicles to service customers plays an important role in the distribution chain. This research domain has thus been widely studied in the literature (see for example, the survey in Ball et al. (95)). However, there is still a lack of modeling approaches that more closely represent real-life conditions. One practical aspect that has seldom been addressed is the time dependency of travel times on the time of the day. Most available models simply assume that the travel times are constant throughout the day. Unfortunately, this assumption is a weak approximation of the real-world conditions where travel times are subject to variations over time. These variations may result from predictable events (e.g., congestion during peak hours) or from unpredictable events like accidents, vehicle breakdowns, and others. Therefore, the optimal solution to a formulation of the problem that assumes constant travel times may be suboptimal or even infeasible for the time-dependent problem (e.g., if time windows are considered).

Time-dependent vehicle routing problems have seldom been addressed because they are harder to model and to solve. These problems can be stated as follows. Let assume that a fixed size fleet of m identical vehicles of fixed capacity is available to service customers (nodes) with fixed demand and that the time horizon is partitioned into p time intervals $T_1, T_2, ...T_p$. Given a network of n nodes, a nxn time-dependent matrix $C(T_k)_{k=1,2,...p} = [c_{ij}(T_k)]$ contains the travel times between each pair of nodes (i, j) when the vehicle departs from node i within time interval T_k. The goal is to find a set of minimum cost vehicle routes that service every customer, such that :

- Each vehicle route originates from and terminates at a fixed depot ;

- Each vehicle services one route and the service point of each request is visited once by exactly one vehicle;
- The capacity of each vehicle is not exceeded and the demand of each customer is satisfied;

Furthermore, each service point i (including the depot) may have its own time window $[e_i, l_i]$, where e_i is the earliest service time and l_i is the latest service time. When the service points have "soft" time windows, a vehicle can arrive before the lower bound or after the upper bound. If the vehicle is too early, it must wait to start its service; if the vehicle is too late, a penalty for lateness is incurred in the objective function.

The problem considered here is motivated from a courier service application found in the local operations of long-distance shipping services, where parcels and envelopes are collected at different customers' locations and brought back to a central office for further processing and shipping. In this case, the service points have "soft" time windows, but each route must start and end within the time window associated with the depot. Also, no capacity constraint is enforced since only small items are transported. The cost to be minimized is a weighted sum of the total travel time over all routes, plus the total lateness over all customers. A time-dependent model for predictable variations in travel times is proposed and analyzed in this context. It is implemented in a static environment where the customer requests are known in advance (i.e., before the routes are constructed), and in a dynamic environment where new customer requests are unveiled as the routes are executed.

The paper is organized as follows. Section 2 presents a brief literature review dedicated to time-dependent vehicle routing and two other problems closely rela-

ted to it, the time-dependent shortest path and the path choice problems. Section 3 presents our time-dependent model and discusses several issues related to it. Section 4 briefly introduces a tabu search heuristic developed by Taillard et al. (97) for a version of the problem with fixed travel times and explains how it was modified to account for time-dependency. Section 5 reports experimental results obtained with the new algorithm in a static context. Section 6 addresses the dynamic version of the problem. Finally, section 7 concludes and proposes future avenues of research.

4.2 Literature review

The only papers that we are aware of in the literature of time-dependent vehicle routing are the ones of Malandraki (89), Malandraki and Daskin (92) and Hill and Benton (92). Malandraki and Daskin (89, 92) examined both the time-dependent vehicle routing problem (TDVRP), and the time-dependent traveling salesman problem (TDTSP) which is a special case of the TDVRP when the fleet size is equal to one. They provided mixed integer linear programming formulations which include time windows, capacities and allow for waiting at a customer location. The travel times were computed using step functions. Nearest-neighbor (greedy) heuristics for the TDTSP and the TDVRP without time windows were proposed, as well as a branch-and-cut algorithm for solving small problems with 10 to 25 nodes.

Hill and Benton (92) considered a time-dependent vehicle routing problem (without time windows) and proposed a model based on time-dependent travel speeds. Computational results were reported on a small example with a single vehicle and five locations. The authors also mentioned the implementation of a simple greedy

heuristic for the multi-vehicle traveling salesman problem with capacity constraints and no time windows for a city with 210 locations. A validation of the model within a commercial courier scheduling package is also mentioned, but no details are provided.

The major weakness of the above models is that they do not satisfy the "first in first out" (FIFO) property. The FIFO property guarantees that if a vehicle leaves a node i for a node j at a given time, any identical vehicle leaving node i for node j at a later time will arrive later at node j (which is common sense). This will be discussed in section 3.

Time-dependency has seldom been addressed in the literature on vehicle routing. However, it has been widely studied in three related problems : the time-dependent traveling salesman problem, the shortest path problem and the path choice problem. They are briefly described below.

Time-dependent shortest path problem

The earliest models which account for time-dependency were developed by the end of the fifties for solving shortest path problems (Ford and Fulkerson (58), Cooke and Halsey (66), Dreyfus (69)). Since then, these problems have been the most widely studied. The goal is to find minimum cost paths from origin nodes to destination nodes, through a network where travel times and costs are time-dependent.

Time-dependent path choice problem

The earliest work that we are aware of is the one by Marguier and Cedar (84) for common bus stops with overlapping routes. In the time-dependent path choice problem, some characteristics of the problem are stochastic (e.g., arrival times, waiting times, travel times, etc.). The goal is to determine, under uncertainty, op-

timal routes among all possible routes between specific origins and destinations, in a network where travel times and costs are time-dependent. The objective function is typically the expected total travel time.

Time-dependent traveling salesman problem (TDTSP)

The earliest papers in the literature related to time-dependent vehicle routing problems appeared in the sixties and were dedicated to the time-dependent traveling salesman problem (Miller et al. (60) and Hadley (64)). The TDTSP constructs a Hamiltonian tour of minimum travel cost over n cities,where the travel cost from city i to city j depends on the time of the day.

According to the properties of the travel time and cost functions, the work related to time-dependent problems can be classified in four main categories. They are briefly presented in the following subsections.

4.2.1 Models based on "simple" travel time and cost functions

Many researchers have used simple rules to integrate time-dependency components in their models. In their decision support system for dispatching and processing customer orders for gasoline, Brown et al. (87) first produce a solution where travel time fluctuations are ignored. Then, the loads for each truck are resequenced "manually" to take into account various factors such as traffic congestion during rush hours, road and weather conditions, etc. Other researchers used multiplier factors to represent variations in travel times (Fisher et al. (82), Hill et al. (88), Rousseau and Roy (88), Shen and Potvin (95)). Clearly, this is just a rough approximation of actual conditions.

4.2.2 Models based on discrete travel time and cost functions

In this kind of formulation, the horizon of interest is "discretized" into small time intervals. The travel time and cost functions for each link are assumed to be step functions of the starting time at the origin node. This scheme is widely used in many time-dependent transportation problems. However, the assumption that travel times vary in discrete jumps is just an approximation of real-world conditions since travel times change continuously over time.

Many researchers used this kind of model to solve time-dependent shortest path problems (Cooke and Halsey (66), Dreyfus (69), Ziliaskopoulos and Mahmassani (93), Ziliaskopoulos (94) and Chabini (96, 97)). This framework was also used to formulate time-dependent traveling salesman problems (Picard and Queyranne (78), Fox et al. (80), Malandraki (89) and Malandraki and Daskin (92)). Finally, Natchigall (93) used discrete transit functions to model a time-dependent path choice problem in a railway context.

The major drawback of using models that are based on discrete travel time and cost functions is that the FIFO property does not hold (see section 3).

4.2.3 Models based on continuous travel time and cost functions

In real-life, travel times vary continuously over time. The work reported in this section is thus aimed at modeling the reality more accurately. In their formulations of time-dependent shortest path problems, Halpern (77) and De Palma et al. (90) calculated the travel times using nonnegative and piecewise linear functions while

Orda and Rom (90) assumed that the travel times were arbitrary. In their models dedicated to time-dependent path choice problems, Hall (86), Hickman and Wilson (95), and Hickman and Bernstein (97) assumed that the travel times are stochastic and time-dependent. Finally, Marguier and Cedar (84) studied the same problem for common bus stops with overlapping routes. Time-dependent distributions were used to represent passengers' waiting times.

Continuous and arbitrary travel time functions seem to be more appropriate to model real-world conditions. Unfortunately, models based on such functions are confronted with the following limitations : **(i)** a lot of simplifying assumptions are often made to make the model more tractable (e.g., differentiability, piecewise linear functions, etc...) ; **(ii)** the functions used are not necessarily related to what is observed in the real world since the performance of an algorithm strongly depends on the characteristics and properties of the functions considered ; **(iii)** in the case of the path choice problem, using continuous time-dependent distributions to represent travel time or waiting time functions results in complicated integrations that are difficult to solve analytically. Furthermore, this kind of formulation often suggests that passengers make their boarding decision using a fairly complicated logic (Hickman and Bernstein (97)).

4.2.4 Models based on Markovian formulations of travel time and cost functions

The only work that we are aware of in this category is the one of Psaraftis and Tsitsiklis (93). The authors investigated the shortest path problem in a stochastic and time-dependent setting. The cost of each arc (i, j) is a known function $f_{ij}(e^i)$ of the state e^i of some environment variable at node i at the time of departure from

node i to node j. Environment variables are mutually independent and governed by a finite state Markov process where state transitions occur in discrete time. The goal is to seek a policy that minimizes the expected total cost on a path between two specific nodes. A dynamic programming algorithm was proposed to solve the problem. However, given that its complexity depends on the number of Markov states at each node, the state space grows quickly with problem size, thus preventing the model from being applied to realistic problem instances.

In section 3, we propose and analyze a model that focuses on travel speed and travel speed variations from one time period to the next.

4.3 A time-dependent travel speed model

4.3.1 Motivation

As mentioned previously, the literature related to time-dependent vehicle routing problems is very scarce. In fact, the only papers that we are aware of in this category are those of Malandraki (89), Malandraki and Daskin (92) and Hill and Benton (92). They were briefly discussed in section 2. The major shortcoming of Malandraki's model, which represents the travel time as a step function of time, is that the FIFO assumption does not hold. To illustrate this, consider figure 4.1 which represents a travel time function on a link (i, j) of length 1.

Based on this function, if the vehicle leaves node i at instant $t_1 = 1$, it will reach node j at instant $t_1' = 4$ (since the travel time is equal to 3 time units for a departure at time 1). However, it will arrive at instant $t_2' = 3$ $(< t_1')$ if it leaves node i at instant $t_2 = 2$ $(> t_1)$, since the travel time is equal to 1 in this case. To overcome this weakness, Malandraki and Daskin (89, 92) suggested to

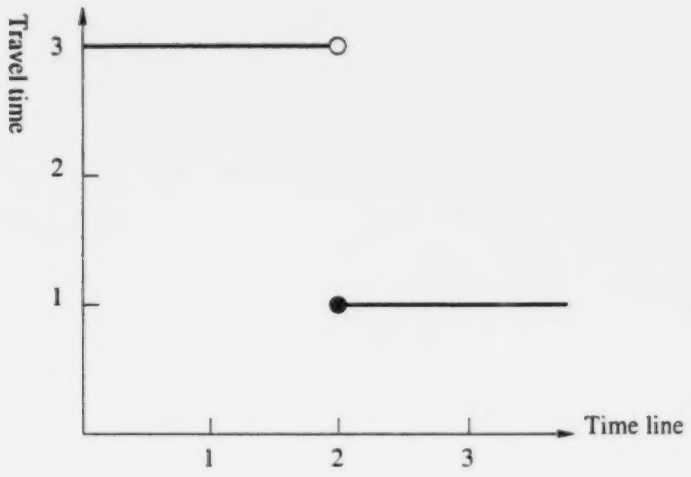

FIG. 4.1 – Travel time function on a link.

allow vehicles to wait at a node to smooth the travel time function. However, this suggestion concerns decreasing step functions only and induces useless waiting at nodes. In real-life applications (e.g., distribution of industrial products) it is hard to convince a dispatcher that it is advantageous to force a driver to wait at a customer location even if he is ready to depart.

We recall that Hill and Benton (92) developed the only model that we are aware of based on time-dependent travel speeds. In this formulation, the travel time on a given link (i, j), starting during time period T, is

$$d_{ij}/r_{ijT}, \text{ with } r_{ijT} = (r_{iT} + r_{jT})/2, \qquad (1)$$

where d_{ij} is the distance between locations i and j, and r_{iT} is the average speed associated with the "area" around location i during time period T. Thus, r_{ijT} is an average travel speed for a trip from i to j that starts during time period T. But since the speed along a given link (i, j) is an average speed based on a single period, the FIFO assumption is not necessarily satisfied either. To illustrate this, consider the arrival time at node j from node i when all nodes have the same speed

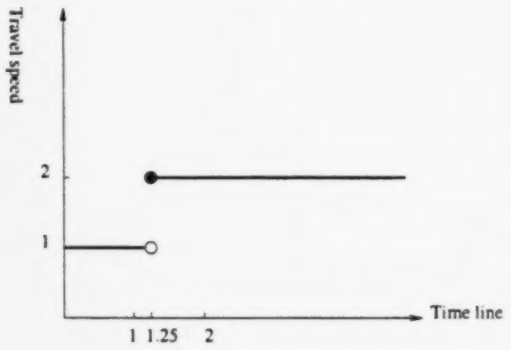

FIG. 4.2 – Travel speed function at a node.

value r_{iT} for a given time period T. Figure 4.2 illustrates the travel speed function for a node.

Assuming that link (i, j) has length 1, if the vehicle leaves node i at instant $t_1 = 1$, it will arrive at node j at instant $t_1' = 1 + 1 = 2$ (since the speed is 1 for a departure at time 1, it will take 1 time unit to travel along a link of length 1). However, if the vehicle departs from node i at instant $t_2 = 1.25$ $(> t_1)$, it will reach node j at instant $t_2' = 1.25 + 0.5 = 1.75$ $(< t_1')$. Since in the latter case, the travel speed is 2 for a departure at time 1.25, it takes only 0.5 time units to travel along a link of length 1.

In fact, associating a time-dependent travel speed or a time-dependent travel time with a link is equivalent since it is always possible to deduce the travel speed from the travel time and conversely for a given travel distance. To better model time-dependency, one has to take into account the adjustment of the travel speed when the vehicle crosses the boundary between two consecutive time periods. In the following, we propose a model that addresses this issue.

4.3.2 The model

In vehicle routing problems and more generally in the transportation field, an important area that remains very challenging is the conception of efficient models to achieve a good trade-off between the implementation requirements and the ability to reflect the complexity of real-world conditions such as fluctuations in travel times. A "natural" and simple way to take time-dependency into account is to work with time-dependent travel speeds and to adjust the speed when the vehicle crosses a boundary between two time periods.

Let assume that the horizon is divided into p time periods $T_1, T_2, ..., T_p$. Given a network of n nodes, a symmetric distance matrix $D = (d_{ij})$ and travel speed matrices $V_T = (v_{ijT})$, $T \in \{T_1, T_2, ..., T_p\}$ are defined. In contrast with the formulation proposed by Hill and Benton (92) where travel speeds are indexed by time periods and nodes, here, the travel speeds are indexed by time periods and arcs. This reduces the computational effort at the cost of more storage (in Hill and Benton (92), we recall that the travel speed on a link (i, j) is calculated as $r_{ijT} = (r_{iT} + r_{jT})/2$). To limit the number of speed values v_{ijT} to estimate, the set of arcs A is partitioned into subsets $(A_c)_{1 \leq c \leq C}$. That is, the travel speed on an arc (i, j) that belongs to a subset (or category) A_c at period T is $v_{ijT} = v_{cT}$, where v_{cT} is the travel speed associated with category A_c and time period T. Consequently, the number of parameters in the model is considerably reduced, especially if C is small. Partitioning the set of arcs into subsets is a reasonable assumption for urban transportation networks since links (routes) can usually be classified into categories based on their physical characteristics (e.g., width, one/two ways, etc...), and their geographical location.

4.3.2.1 Changing travel speeds

The main point in our model is that we do not assume a constant speed over the entire length of a link. Rather, the speed changes when the boundary between two consecutive time periods is crossed. This is illustrated in figure 4.3 where a route for one vehicle is considered. The horizon is "discretized" into three time periods $(T_j)_{1 \leq j \leq 3}$, with a different speed associated with each period. The vehicle leaves service point i at time $t_0 \in T_1$ and travels at speed v_{1T_1} until it reaches point i_1 at the boundary between periods T_1 and T_2. From there, the vehicle travels at a faster speed v_{1T_2} until it reaches point i_2 at the boundary between T_2 and T_3. Finally, it travels at speed v_{1T_3} for the remainder of its trip to service point j.

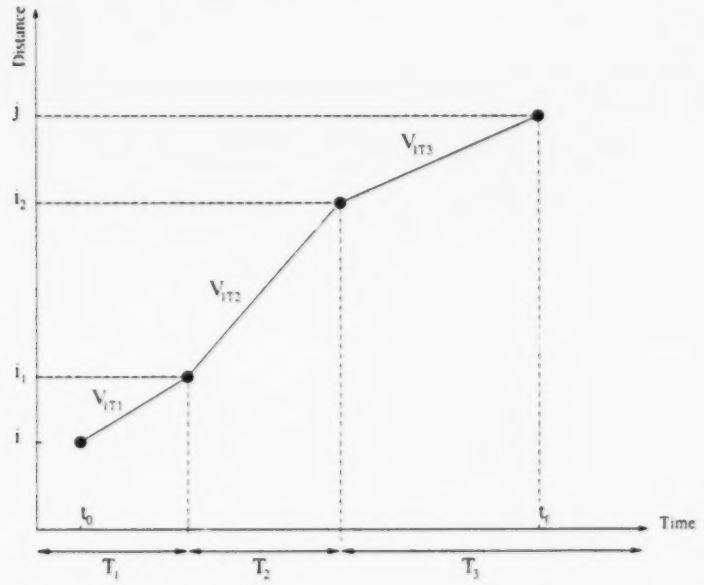

FIG. 4.3 – Changing travel speed over time.

1. set t to t_0,
   set d to d_{ij},
   set t' to $t + (d/v_{cT_k})$.
2. while $(t' > \bar{t}_k)$ do
   2.1 $d \leftarrow d - v_{cT_k}(\bar{t}_k - t)$.
   2.2 $t \leftarrow \bar{t}_k$
   2.3 $t' \leftarrow t + (d/v_{cT_{k+1}})$.
   2.4 $k \leftarrow k + 1$.
3. return $(t' - t_0)$.

FIG. 4.4 – Travel time calculation procedure.

4.3.2.2 Travel time calculation

Figure 4.4 describes the procedure for calculating the travel time between any pair of nodes i and j. We suppose that the vehicle leaves node i at $t_0 \in T_k =]\underline{t}_k, \bar{t}_k]$ and that link (i, j) belongs to category c, $1 \leq c \leq C$. It is assumed that d_{ij} is the distance between i and j, and v_{cT_k} is the travel speed associated with category c and time period T_k. Also, t denotes the current time and t' denotes the arrival time.

4.3.2.3 Characteristics of the travel time function

In our model, the travel speed v_{cT} is a step function of the time of day. Therefore, the travel time is a piecewise continuous function over time that is simple and easy to evaluate beside being a "natural" way to estimate travel times in real-world conditions. Figure 4.5 gives an example of the travel speed function and its associated travel time function for a link of length 1. Given the travel speed function, the travel time function can be derived as follows. Since the speed is equal to 1 for a departure at instant 0, it takes 1 time unit to travel along a link of length 1. However, for any departing time t_1 such that $0 < t_1 < 1$, the vehicle will

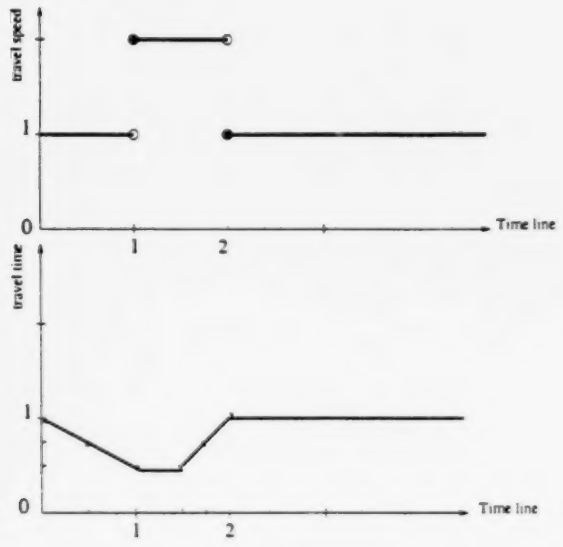

FIG. 4.5 – An example of travel speed and travel time functions.

travel at speed 2 on a part of the link. Hence, the travel time function decreases in the interval $]0, 1[$. During the interval $[1, 1.5]$, the vehicle can travel at speed 2 along the entire link. Therefore, the travel time along this link is constant at 0.5 time units. For any departing time t_1 such that $1.5 < t_1 < 2$, the vehicle will travel at speed 1 on a part of the link. Hence, the travel time function increases in the interval $]1.5, 2[$. After $t_1 = 2$, the travel time remains equal to 1 time unit since the travel speed function is constant at 1.

We are aware that travel speeds also change continuously over time, however using step functions to compute travel speeds is a more reasonable assumption than for travel times. That is, when the boundary between two consecutive time periods is crossed, the speed will typically change much faster than the travel time on a link.

Due to the travel speed changes, the travel time function proposed in this model satisfies the FIFO property. That is, leaving a node earlier guarantees that one will

arrive earlier at his destination. Consider the previous example given in figure 4.5. A vehicle that is ready to leave the link before $t_1 = 1$ has no incentive to wait even if its speed will increase afterward. Actually, during this waiting time, the vehicle could have used the available speed to get closer to its destination until the time of speed change. At this point, the faster speed could be used to terminate the trip. Hence, the FIFO property precludes our model from inducing useless waiting time as it is the case in previous formulations.

This approach has been implemented within a parallel tabu search developed by Taillard et al. (97) for the fixed travel time version of the problem. The next section will briefly introduce the original algorithm and then, explain how it was modified to cope with time-dependent travel times.

4.4 A parallel tabu search algorithm

4.4.1 The original algorithm

The algorithm developed in Taillard et al. (97) is a parallel tabu search heuristic with an adaptive memory. Tabu search is an iterative local search technique that starts from some initial solution. At each iteration, a neighborhood is generated around the current solution and the best solution in this neighborhood becomes the new current solution (even if it does not provide an improvement). By allowing a degradation of the objective, it is possible to escape from bad local optima, as opposed to pure descent methods. The interested reader will find more details about this approach in Glover and Laguna (97).

The algorithm developed in Taillard et al. (97) can be summarized as follows :

– Construct I different initial solutions with a stochastic insertion heuristic

(where the choice of the next customer to be inserted is randomized).

- Apply tabu search to each solution and store the resulting routes in an adaptive memory.
- While a stopping criterion is not met do :
  - Use the routes stored in the adaptive memory to construct an initial solution.
  - Decompose the problem into subproblems obtained through a geographical decomposition procedure.
  - Apply tabu search to each subproblem.
  - Add the routes of the resulting solution in the adaptive memory.

The objective is to minimize a weighted sum of total distance traveled and total lateness over all customers. The procedure for generating the neighborhood of the current solution is the CROSS exchange. Basically, two segments of variable lengths are taken from two different routes and swapped. Figure 4.6 illustrates this procedure.

To speed up the algorithm, a parallel implementation on a network of workstations was developed (Taillard et al. (97)). The parallelization of the procedure was achieved at two levels :

(1) Different tabu search threads run in parallel, each of them starting from a different initial solution.

(2) Within each search thread, many tabu searches run independently on the subproblems obtained through the decomposition procedure.

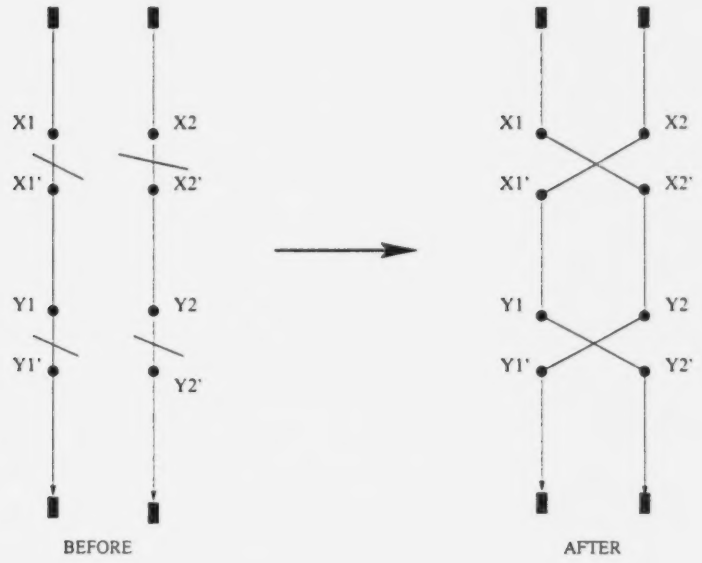

FIG. 4.6 – A CROSS exchange.

4.4.2 Implementing the time-dependent model

Implementing the time-dependent model mostly impacts the evaluation of a new solution after a CROSS exchange. In the following, we explain how the original algorithm was modified to cope with time-dependency.

4.4.2.1 Objective function

To evaluate a given solution, the total distance traveled is replaced by the total travel time in the objective function.

4.4.2.2 Neighborhood evaluation

The evaluation of a move, as implemented in the original algorithm cannot be directly applied to our problem. In the following, the original evaluation process is briefly described, then its adaptation to the time-dependent problem is discussed

in detail.

The original procedure

As mentioned before, the original algorithm uses CROSS exchanges to generate the neighborhood. The evaluation of such moves is based on the difference between the value of the neighboring solution and the value of the current solution. Namely, $\delta f = \delta d + \alpha * \delta l$, where δd and δl are respectively, the modification to the total distance and the modification to the total lateness of the solution, and α is a constant parameter. The evaluation of δd is done in constant time by simply subtracting the total length of the edges that are removed from the solution and by adding the total length of the edges that are introduced into the solution. The modification to the total lateness of a solution is the sum of the modifications to the total lateness incurred by both routes involved in the CROSS exchange. Unlike δd, this modification cannot be evaluated exactly in constant time because any additional lateness at a given customer location must be propagated along the route. To reduce the complexity of the calculation, an approximate evaluation procedure is used to evaluate each neighboring solution. Then, the M best solutions according to this approximation are evaluated exactly and the best solution is selected.

To illustrate the approximate evaluation procedure, the contribution δl of the new route servicing customers X_1, X_2', Y_2 and Y_1' in fig. 4.6 is considered. The first part of this contribution is evaluated exactly in constant time by propagating the lateness at customer X_2' along the route segment $X_2' - Y_2$ (which is limited to at most 7 customers, see Taillard et al. (97)). The second part of δl is assessed using an approximation function associated with customer Y_1'. To construct the approximation function associated with each customer i, the variation δl_i of total lateness

on the route of customer i is calculated exactly for a number of delays δb_i to the start of service at this customer. Then, a piecewise linear function is produced by interpolation. The interested reader will find further details in Taillard et al. (97).

Adaptation to the time-dependent model

In the proposed time-dependent model, the difference between the value of the neighboring solution and the current solution is $\delta f = \delta tr + \alpha.\delta l$, where δtr is the modification to the total travel time of the solution.

(a) Lateness

Due to the time-dependency component, the exact evaluation of lateness is more computationally expensive than in the original problem. Thereby, using an approximate evaluation procedure is even more important in our case. In the following, the adaptation of the original approximation procedure is illustrated in the case of the new route servicing customers X_1, X_2', Y_2 and Y_1' in figure 4.6.

(i) The contribution of the route segment $X_2' - Y_2$ to the total lateness is assessed exactly as in the original algorithm. However, the time-dependent travel time calculation described in subsection 4.3.2 is used. This provides the value $\delta b_{Y_1'}$ at Y_1'.

(ii) The contribution of the remainder of the route is evaluated approximately as in the original procedure. That is, an approximate function is associated with each customer i, based on an exact evaluation of δl_i for a few values of δb_i, using again the time-dependent travel time function previously introduced. Now, suppose that we search for the value of $\delta l_{Y_1'}$ induced by $\delta b_{Y_1'} = z$ with $z_j \leq z \leq z_{j+1}$, where z_j and z_{j+1} are two consecutive values of $\delta b_{Y_1'}$ for which $\delta l_{Y_1'}$ is known exactly (i.e., these values have been used to construct the linear interpolation). In figure 4.7, we assume that $\delta l(z)$, δl_j and δl_{j+1} are the values of $\delta l_{Y_1'}$ associated with z, z_j and z_{j+1} respectively. The interpolation is such that $\delta l_j \leq \delta l(z) \leq \delta l_{j+1}$, which makes sense

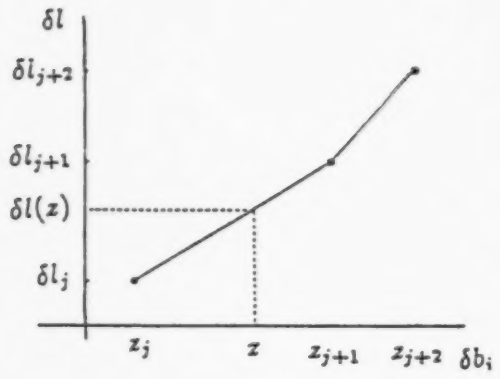

FIG. 4.7 – Approximate evaluation of lateness.

since the travel time function satisfies the FIFO rule (that is, the exact variation of lateness $\delta l(z)^*$ should also satisfy $\delta l_j \leq \delta l(z)^* \leq \delta l_{j+1}$).

(b) Travel time

As opposed to the variation of the total distance used in the original algorithm, assessing the modification to the total travel time of a given solution cannot be achieved easily. To illustrate this purpose, consider the two routes in figure 4.6. Subtracting the total length of the removed edges and adding the total length of the new edges, was sufficient to evaluate the total distance variation. However, in the time-dependent context, it may well happen that the total travel time over route segment $X_1' - Y_1$ or (and) over route segment $X_2' - Y_2$ change after executing the move (i.e., the total travel time over segment $X_1' - Y_1$ before the move depends on the arrival time to X_1' from X_1 ; after the move, it depends on the arrival time to X_1' from X_2, which may be different). To overcome this difficulty, an approximate evaluation is used, similar to the one presented in (a).

At the end, the M best moves according to the approximation, are evaluated exactly using the time-dependent travel time function, and the best move is executed to obtain the new current solution. M is a parameter that needs to be

adjusted, since the better the approximation is, the lower the value of M needs to be.

To examine the potential savings that may be obtained through the use of the proposed time-dependent approach, several tests were conducted. The next section reports computational results obtained with the new algorithm.

4.5 Computational results

In this section, we report experimental results obtained with the time-dependent algorithm. First, we describe the test problems and then we report numerical results.

4.5.1 Test problems

Our model was validated on Solomon's 100-customer Euclidean problems (see Solomon(87)). In these problems, customer locations are generated within a $[0, 100]^2$ square. Six different sets of problems are defined, namely C_1, C_2, R_1, R_2, RC_1 and RC_2. The customers are uniformly distributed in the problems of type R, clustered in groups in the problems of type C and mixed in the problems of type RC. In the problems of type 1, only a few customers can be serviced on each route due to a narrow time window at the depot, as opposed to problems of type 2 where each route may have many customers. The travel times were calculated using a 3x3 time-dependent travel speed matrix $(v_{cT})_{1\leq c\leq 3, 1\leq T\leq 3}$, where each row corresponds to a category of arc and each column to a time period. Within the scheduling horizon, the first and third periods stand for the morning and evening rush hours, respectively. The second period corresponds to the middle of the day, when the traffic density is lower.

Entries of the travel speed matrix were adjusted to create three different types of scenarios. For each scenario, the travel speeds in the morning and evening rush hours were obtained by dividing the travel speeds in the middle of the day by a factor α. In scenarios 1, 2 and 3, α was set to 1.5, 2 and 4 respectively. Hence, scenario 3 is the one with the highest degree of time-dependency, while scenario 1 is the one with the lowest. The travel speed matrix for the three scenarios are reported in tables 1, 2 and 3, respectively. The average speed in each matrix is approximatively 1 so the "average" difficulty is the same as in Solomon's original problems.

0.54	0.81	0.54
0.81	1.22	0.81
1.22	1.82	1.22

TAB. 4.1 – *Travel speed matrix in scenario 1*

0.33	0.67	0.33
0.67	1.33	0.67
1.33	2.67	1.33

TAB. 4.2 – *Travel speed matrix in scenario 2*

0.12	0.46	0.12
0.46	1.92	0.46
0.96	3.84	0.96

TAB. 4.3 – *Travel speed matrix in scenario 3*

4.5.2 Experiments

The experiments reported in this section were performed on a network of 9 SUN UltraSparc-IIi workstations (300 MHz). As mentioned previously, the objective is to minimize the sum of total travel time and total lateness over all customers. In these experiments, the fleet size was set to the number of routes in the best solution reported in the literature for each problem. The value of the parameters were kept as in the original algorithm (see Gendreau et al. (99)). With respect to parameter M, which corresponds to the number of best moves (according to the approximation) which are evaluated exactly, several values were tested.

For each scenario, each problem is solved assuming time-dependent travel speeds, and then assuming constant speeds (i.e., for each category of arcs, the average speed is taken over the three time periods). The two solutions are then compared, using the time-dependent travel speed matrix, to evaluate what is gained by explicitly considering variations in travel times over the day rather than using the approximation represented by the average.

4.5.2.1 Preliminary tests

Scenario 1 was first considered for the preliminary tests. The intent was to find the best value for parameter M. The preliminary experiments were conducted over a small sample of problems selected in each of the six classes C_1, C_2, R_1, R_2, RC_1, and RC_2. Several values had to be tested for every problem in the sample. Thereby, a significant amount of computation time is required if the size of the sample is too large. On the other hand, this size has to be large enough to get a good calibration. In our experiments, the size of the sample was set to 4. Table 4 reports results obtained with the time-dependent travel speed matrix of scenario 1, and

the values 4, 8, 16, 24, 32 and 40 for parameter M. The four numbers in each entry are the travel time, lateness, objective value and running time (in minutes) respectively, for each problem set. In these experiments, the algorithm stops after 50 restarts from the adaptive memory (see subsection 4.1). As expected, increasing the value of M increases the running time and decreases the objective value. However, the resulting increase in running time is small when compared to the total running time. In fact, since the neighborhood of a given solution is large, the algorithm spends much more time at evaluating the neighboring solutions, even approximatively, than it does at evaluating a few solutions exactly. Table 4 was used to determine the value for M that achieves the best trade-off between execution time and solution quality. We observed that $M = 32$ and 40 lead to approximatively the same results while the execution time for $M = 32$ is slightly smaller. Hence, the value 32 was retained for further testing.

To quantify the effort of using the proposed time-dependent model instead of the original algorithm, we computed the amount of time consumed by one iteration of each of the two algorithms. Preliminary results have shown that using the time-dependent model leads to a very small increase in computational time (about 15.86 seconds against 15.29 seconds for problems of classes C_1, R_1 and RC_1, and about 89.78 seconds against 86.36 seconds for problems of classes C_2, R_2 and RC_2).

4.5.2.2 Numerical results

Table 5 compares, for scenarios 1, 2 and 3, the solutions obtained with time-dependent travel speeds and the solutions obtained with constant speeds over all problems in each category. The four numbers in each entry are the number of infeasible solutions, the average travel time, total lateness and objective value,

respectively, for each problem class. The row "Overall" contains averages taken over the entire set of 56 test problems. It is worth noting that a solution obtained with constant speeds may well be infeasible in the time-dependent context (i.e., the upper bound of the time window at the depot may well be exceeded). This is what the first number in each entry indicates. The results show that a significant number of solutions obtained with constant speeds are infeasible in the time-dependent context and that this number increases with the degree of time-dependency. Also, the use of time-dependent travel speeds considerably improves the objective value. This improvement is observed under the three scenarios, for all problem classes. More precisely :

Scenario 1 : An improvement is observed in all classes of problems (apart from R_2). The reduction ranges from 1.0% to 5.0%. In the case of R_2, the two models lead to approximatively the same results.

Scenario 2 : As expected, the model performs better under scenario 2 where the degree of time-dependency is higher. In fact, the results show an improvement in all problem classes. The reduction ranges from 2.0% to 12.5%.

Scenario 3 : In this scenario, the results are the most impressive. This is not surprising since the degree of time dependency is the highest. An improvement is observed in all problem classes. The reduction in the objective value ranges from 9.2% to 18.0%.

4.6 Dynamic problem

The proposed model was also tested in a dynamic environment. In this section, the dynamic time-dependent vehicle routing problem is first introduced. Then, we explain how the algorithm of Taillard et al. (97) for the static problem with fixed

travel times was adapted to the dynamic case (Gendreau et al. (99)). We then describe how we modified the later algorithm to cope with time-dependent travel times in a dynamic environment. Finally, computational results are reported.

4.6.1 Problem definition

As opposed to the static problem, the number of service requests are not known completely ahead of time, but are rather dynamically revealed as time goes by. As illustrated in figure 4.8, in a dynamic environment, a vehicle route can be divided into three parts at any instant t :

- completed movements which form the part of the route already executed. Thereby, this part cannot be modified anymore ;
- current movement of the vehicle toward its current destination ;
- planned movements which constitute the portion of the route not yet executed by the vehicle (planned route).

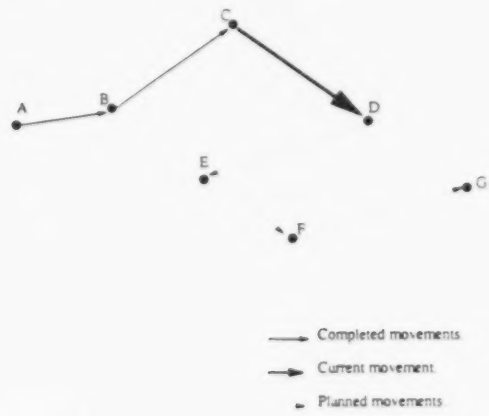

FIG. 4.8 – A vehicle route in a dynamic setting.

Whenever a new request unfolds at instant t, the problem is to assign this request to a particular vehicle and incorporate it into its planned route at minimum cost.

4.6.2 Original dynamic algorithm (Gendreau et al. (99))

In a dynamic setting, at any instant t, a solution is a set of planned routes, each beginning with the current destination of the associated vehicle. The major modification for adapting the static algorithm to the dynamic case was related to the management of new incoming events. In Gendreau et al. (99), a new event may be of two types : the occurrence of a new service request, or the completion of service at a customer (given that the next destination must be indicated to the driver). Whenever any of these events occurs, the tabu search threads are interrupted. Then, after an appropriate update of the adaptive memory to reflect the current state of the world, the search is restarted with new solutions constructed from the updated memory. The tabu search is thus used to improve the current (static) solution between the occurrence of new events.

4.6.3 Time-dependent algorithm in a dynamic environment

In addition to the modifications already discussed in subsection 4.2, another modification is required in the dynamic case. In the original algorithm, a least commitment strategy is considered. That is, if there is some time flexibility at the vehicle's next destination, the vehicle waits at its current location rather than its next destination. The intent is to allow the vehicle to service a new request that may appear in the vicinity of its current location. Hence, one has to determine a value for the departure time that allows the vehicle to reach its next customer no earlier than the lower bound of its time window.

When time-dependency is taken into account, the adjustment of the vehicle departure time is more complicated, because the travel time between a given pair of locations depends on the departure time from the origin. Hence, one has to take

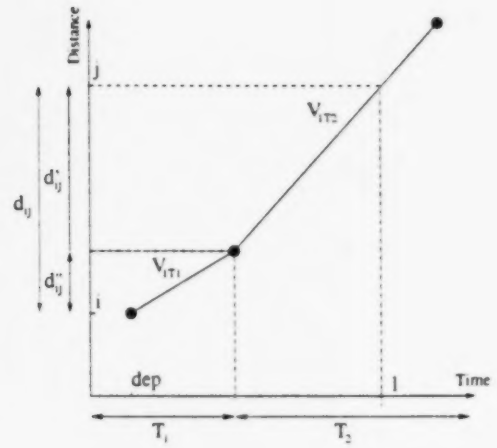

FIG. 4.9 – Finding the departure time for a vehicle in a dynamic environment.

into account the change that occurs in travel speed when the boundary between two consecutive time periods is crossed. This problem is solved through a backward recursive procedure (in contrast with the forward procedure used to compute the travel times between customers).

In figure 4.9, a route for one vehicle is considered. The horizon is "discretized" into two time periods $(T_k)_{1 \leq k \leq 2}$. In this figure, i is the current customer and j is the next customer to be serviced, with arc (i, j) belonging to category 1. We also suppose that the time of departure *dep* belongs to time period $T_1 =]\underline{t}_1, \bar{t}_1]$ and that l, the lower bound of the time window at customer j, belongs to time period $T_2 =]\underline{t}_2, \bar{t}_2]$, with $\bar{t}_1 = \underline{t}_2$. With the least commitment strategy, the departure time *dep* from node i is calculated as follows. First, we evaluate the distance traveled during time period T_2 which is $d'_{ij} = v_{1T_2}(l - \underline{t}_2)$. Since the remaining distance is $d''_{ij} = d_{ij} - d'_{ij}$, we obtain $dep = \bar{t}_1 - d''_{ij}/v_{1T_1}$.

A pseudo-code for this backward recursive procedure is provided in figure 4.10. For a given vehicle, it calculates the departure time from its current customer i to

arrive at the lower bound l of the time window associated with its next destination j, where $l \in T_k =]\underline{t}_k, \bar{t}_k]$. In this figure, t is a time variable which is updated at each iteration and *dep* corresponds to the departure time. It is also assumed that arc (i, j) belongs to category c, $1 \leq c \leq C$.

The modified algorithm was used to assess the proposed time-dependent model in a dynamic context. Details about the experiments are reported in the next subsection.

1. set t to l,
   set d to d_{ij},
   set *dep* to $l - (d/v_{cT_k})$.
2. while $(dep < \underline{t}_k)$ do
   2.1 $d \leftarrow d - v_{cT_k}(t - \underline{t}_k)$.
   2.2 $t \leftarrow \underline{t}_k$
   2.3 $dep \leftarrow t - (d/v_{cT_{k-1}})$.
   2.4 $k \leftarrow k - 1$.
3. return (dep).

FIG. 4.10 – Departure time calculation procedure in a dynamic environment.

4.6.4 Computational results

4.6.4.1 Simulation framework

Simulations were performed to validate our model in a dynamic setting. As in the static case, data were taken from Solomon's 100-customer Euclidean problems (Solomon (87)). The same six classes of problems were considered, namely C_1, C_2, R_1, R_2, RC_1 and RC_2. The set of requests was divided into two subsets. The first subset contains requests that are assumed to be known at the start of the day (50% of the entire requests in our implementation). The second subset contains requests that are received in real-time. Using minutes as time units for

time-related data in Solomon's files, the time horizon was set to 15 minutes. This leads to about 3 requests per minute. The interested reader will find more details about the simulator in Gendreau et al. (99).

4.6.4.2 Numerical results

The experiments were conducted within the same framework used in the static case (see section 5). Hence, the three scenarios previously considered were examined in the dynamic setting, using the same parameter values as those used in the static environment. In the dynamic context, some requests may be rejected when they occur because no feasible insertion place is found in the current routes. This phenomenon is rather marginal, but a few solutions may contain only 98 or 99 customers (rather than 100 customers). Table 6 compares, for scenarios 1, 2 and 3, the solutions obtained with time-dependent travel speeds to those obtained with constant speeds (the same as in the static case). The four numbers in each entry are the number of infeasible solutions, the average travel time, lateness and objective value, respectively, for each problem class.

In the following, we summarize the main findings for each scenario.

Scenario 1 : For all problem classes, the time-dependent model leads to substantial improvements to the objective value, ranging from 4.0% to 13.2%. A total of 18 solutions produced with constant travel times are now infeasible in the time-dependent context.

Scenario 2 : As expected, the time-dependent model leads to larger improvements in the objective value with regard to scenario 1, given the higher degree of time-dependency. The improvement now ranges from 21% to 75%. Furthermore, 32 solutions obtained with constant travel times are infeasible (i.e., more than half

of the solutions).

Scenario 3 : The same general trends are observed with scenario 3. The improvements range from 62.5% to 78.0% and almost all solutions (except four) obtained with constant travel times are infeasible in the time-dependent context.

4.7 Conclusion

Time-dependent vehicle routing is still a very challenging area that needs to be explored, since it provides a more accurate way to model real urban problems. This paper proposed a time-dependent model for a vehicle routing problem with time windows, based on time-dependent travel speeds, which satisfies the FIFO assumption. Characteristics of the model were addressed and discussed. Then, experiments were performed to evaluate the model in a static and a dynamic environment. The results show that the time-dependent model provides a very significant improvement over the model with fixed travel times, indicating the usefulness of exploiting additional information about the problem. Future work will now be aimed at trying to exploit probabilistic information about the future to make better dispatching decisions.

Problem set	$M = 4$	$M = 8$	$M = 16$	$M = 24$	$M = 32$	$M = 40$
C_1	905.188 [a]	856.708	831.53	829.965	827.225	833.62
4	31.765 [b]	27.645	25.272	27.687	25.272	25.272
problems	936.952 [c]	884.352	856.802	857.655	852.498	858.635
	27.833 [d]	28.039	28.962	29.284	29.304	29.508
R_1	1023.62	997.88	980.392	972.38	973.195	963.225
4	59.432	51.59	46.91	47.537	37.767	52.007
problems	1083.05	1049.47	1027.30	1019.92	1010.96	1015.23
	23.579	23.65	24.0625	24.632	24.95	24.929
RC_1	1090.54	1064.38	1039.40	1036.74	1030.57	1030.68
4	49.517	28.002	40.975	38.935	37.225	40.272
problems	1140.06	1092.38	1080.37	1075.08	1067.79	1070.95
	22.602	22.333	22.983	23.404	23.45	23.579
C_2	836.25	758.518	785.675	763.92	759.705	737.045
4	25.34	35.037	35.702	34.667	18.562	39.90
problems	861.59	793.555	821.378	798.588	778.268	776.945
	11.033	11.03	12.201	12.218	12.65	12.921
R_2	996.258	1020.79	969.792	940.088	943.03	941.18
4	68.365	42.262	52.767	22.342	20.62	16.725
problems	1064.62	1063.05	1022.56	962.43	963.65	957.905
	15.891	19.412	23.596	28.271	29.27	29.133
RC_2	1162.63	1155.73	1087.93	1067.97	1015.51	1035.16
4	73.742	39.86	10.247	22.437	29.47	9.362
problems	1236.37	1195.59	1098.18	1090.41	1044.98	1044.522
	10.912	12.558	16.783	16.515	16.867	16.903

[a] Travel time.

[b] Lateness.

[c] Objective value.

[d] Running time (in minutes).

TAB. 4.4 – *Searching for the best value for parameter M*

Problem set	*Scenario1*		*Scenario2*		*Scenario3*	
	Time – dep. speed	*Const. speed*	*Time – dep. speed*	*Const. speed*	*Time – dep. speed*	*Const. speed*
C_1	0 [a]	0	0	0	0	2
9	818.57 [b]	836.878	836.787	857.241	984.556	1044.23
problems	21.514 [c]	15.764	112.553	109.45	399.28	637.99
	840.084 [d]	852.642	949.34	966.691	1383.84	1682.22
R_1	0	0	0	2	0	5
12	935.188	956.213	778.801	816.498	710.076	798.768
problems	26.461	25.655	36.395	33.692	48.907	51.53
	961.648	981.868	815.197	850.19	758.983	850.298
RC_1	0	0	0	1	0	1
8	1038.44	1063.02	873.741	908.692	825.97	900.741
problems	30.927	32.776	26.756	27.634	22.396	33.524
	1069.36	1095.79	900.498	936.33	848.366	934.265
C_2	0	0	0	2	0	6
8	659.221	699.714	764.091	810.00	1250.28	1119.10
problems	13.361	6.507	80.852	154.67	564.28	1038.26
	672.582	706.221	844.944	964.676	1814.56	2157.36
R_2	0	0	0	1	0	4
11	939.255	934.775	861.763	978.285	782.594	873.669
problems	10.698	9.18	8.100	16.581	18.465	40.546
	949.954	943.955	869.864	994.866	801.059	914.215
RC_2	0	1	0	0	0	4
8	1078.17	1081.59	988.124	1032.44	979.758	1034.29
problems	28.998	35.212	22.352	6.039	23.917	63.425
	1107.17	1116.803	1010.48	1038.48	1003.68	1097.71
Overall	0	0.143	0	1.054	0	3.750
56	919.612	929.349	859.806	897.917	900.687	946.905
problems	21.645	20.476	61.623	54.939	165.504	283.713
	941.257	949.825	921.430	952.855	1066.193	1230.618

[a]Number of infeasible solutions.

[b]Travel time.

[c]Lateness.

[d]Objective value.

TAB. 4.5 – *Comparison of time-dependent and constant speeds under the three scenarios in a static setting (with $M = 32$)*

Problem set	*Scenario1*		*Scenario2*		*Scenario3*	
	Time − dep. speed	*Const. speed*	*Time − dep. speed*	*Const. speed*	*Time − dep. speed*	*Const. speed*
C_1	0 [a]	0	0	2	0	9
9	895.718 [b]	947.766	949.783	1080.75	1208.54	2058.99
problems	40.21 [c]	74.135	129.74	367.248	473.072	3190.72
	935.928 [d]	1021.90	1142.52	1447.99	1681.61	5249.71
R_1	0	7	0	10	0	12
12	1069.36	1199.90	931.956	1271.13	875.012	1506.57
problems	92.531	126.793	62.461	395.985	496.492	1490.15
	1161.89	1326.69	994.418	1667.12	924.661	2996.72
RC_1	0	5	0	8	0	8
8	1168.49	1288.60	1023.82	1364.41	974.794	1485.78
problems	132.084	210.266	69.048	595.28	47.77	1238.44
	1300.58	1498.87	1092.87	1959.69	1022.56	2724.22
C_2	0	2	0	6	0	8
8	712.542	696.881	884.101	751.332	1103.11	1184.45
problems	15.751	61.495	44.102	250.248	516.088	1660.43
	728.294	758.376	928.204	1001.58	1619.19	2844.87
R_2	0	3	0	3	0	9
11	1027.61	1086.55	971.037	1242.67	973.65	1520.63
problems	16.396	74.951	12.954	480.605	23.637	2934.54
	1044.00	1161.50	983.992	1723.27	997.287	4455.17
RC_2	0	1	0	3	0	6
8	1196.53	1253.62	1068.67	1342.52	1010.20	1783.80
problems	24.68	40.628	16.956	209.471	31.931	1779.71
	1221.21	1294.25	1085.62	1552.00	1042.14	3563.51
Overall	0	3.232	0	5.482	0	8.928
56	1014.60	1085.599	968.315	1184.210	1014.14	1588.731
problems	54.156	98.434	55.366	388.995	272.17	2076.908
	1068.76	1184.033	1033.80	1573.205	1286.31	3665.639

[a] Nb. of infeasible solutions.
[b] Travel time.
[c] Lateness.
[d] Objective value.

TAB. 4.6 – *Comparison of time-dependent and constant speeds under the three scenarios in a dynamic setting (with $M = 32$)*

Bibliographie

[1] **M.O.Ball, T.L. Magnanti, C.L. Monna and G.L. Nemhauser (eds),** *Network Routing*, Handbooks in Operations Research and Management Science, Volume 8, North-Holland : Amsterdam, 1995.

[2] **G.G. Brown, C. J. Ellis, G. Lenn, W. Graves, David Ronen,** *"Real Time, Wide Area Dispatch of Mobil Tank Trucks"* , Interfaces **17**, 107-120 (1987).

[3] **I. Chabini,** *"Fastest Routes in Temporal Networks Revisited"*, Presented at *Optimization days*, Montreal, May 1966.

[4] **I. Chabini,** *"A New Algorithm for Shortest Paths in Discrete Dynamic Networks"*, Proc. 8th IFAC Symposium on Transportation Systems, Chania, Greece, 551-556 (1997).

[5] **K.L. Cooke and E. Halsey,** *"The Shortest Route Through a Network with Time-Dependent Internodal Transit Times"*, Journal of Math. Anal. Appl. **14**, 492-498 (1966).

[6] **A. De Palma, P. Hansen, M. Labbé,** *"Commuters' Paths with Penalties for Early or Late Arrival Time"*, Transportation Science **24 (4)**, 276-286 (1990).

[7] **S.E. Dreyfus,** *"An Appraisal of Some Shortest Path Algorithms"*, Operations Research **17**, 395-412 (1969).

[8] **M.L. Fisher, A.J. Greenfield, R. Jaikuman and J.T. Lester,** *"A Computerized Vehicle Routing Application"*, Interfaces **12 (4)**, 42-52 (1982).

[9] **L.R. Ford Jr. and D.R. Fulkerson,** *"Constructing Maximal Dynamic Flows from Static Flows"*, Operations Research **6**, 419-433 (1958).

[10] **K.R. Fox, B. Gavish, S.C. Graves,** *"An n-Constraint Formulation of the (Time-Dependent) Traveling Salesman Problem"*, Operations Research **28 (4)**, 1018-1021 (1980).

[11] **M. Gendreau, F. Guertin, J.Y. Potvin and É. Taillard,** *"Parallel Tabu Search for Real-Time Vehicle Routing and Dispatching"*, Transportation Science **33 (4)**, 381-390 (1999).

[12] **F. Glover and M. Laguna,** *Tabu Search*, Kluwer, 1997.

[13] **G. Hadley,** *"Nonlinear and Dynamic Programming"*, Addison-Wesley Publishing Co, Inc., Reading, Mass., 1964.

[14] **R.W. Hall,** *"The Fastest Path through a Network with Random Time-Dependent Travel Times"*, Transportation Science **20 (3)**, 182-188 (1986).

[15] **J. Halpern,** *"The Shortest-Route with Time-Dependent Length of Edges and Limited Delay Possibilities in Nodes"*, Zeitschift fuer Operations Research **21**, 117-124 (1977).

[16] **M.D. Hickman, D.H. Bernstein,** *"Transit Service and Path Choice Models in Stochastic and Time-Dependent Networks"*, Transportation Science **31 (2)**, 129-146 (1997).

[17] **M.D. Hickman and N.H.M. Wilson,** *"Passenger Travel Time and Path Choice Implications of Real-Time Transit Information"*, Transpn. Res.-C **3 (4)**, 211-226 (1995).

[18] **A.V. Hill and W.C. Benton,** *"Modeling Intra-City Time-Dependent Travel Speeds for Vehicle Scheduling Problems"*, J. Opl. Res. Soc. **43 (4)**, 343-351 (1992).

[19] **A. Hill, V. Mabert and D. Montgomory,** *"A Decision Support System for the Courier Vehicle Scheduling Problem"*, Omega Int. J. of Mgmt Sci. **16**, 333-345 (1988).

[20] **C. Malandraki,** *"Time Dependent Vehicle Routing Problems : Formulations, Solution Algorithms and Computations Experiments"*, Ph.D. dissertation, Northwestern University, Evanston, III., 1989.

[21] **C. Malandraki, M.s. Daskin,** *"Time Dependent Vehicle Routing Problems : Formulations, Properties and Heuristic Algorithms"*, Transportation Science **26 (3)**, 185-200 (1992).

[22] **P.H.J. Marguier and A. Cedar,** *"Passenger Waiting Strategies for Overlapping Bus Routes"*, Transportation Science **18 (3)**, 207-230 (1984).

[23] **C.E. Miller, A.W. Tucker and R.A. Zemlin,** *"Integer Programming Formulation on Traveling Salesman Problems"*, J. Assoc. Comput. Mach. **7**, 326-329 (1960).

[24] **K. Nachtigall,** *"Time-Dependent Shortest-Path Problems with Applications to Railway Networks"*, European Journal of Operational Research **83**, 154-166 (1995).

[25] **A. Orda and R. Rom,** *"Shortest-Path and Minimum-Delay Algorithms in Networks with Time-Dependent Edge-Length"*, J. ACM **37**, 607-625 (1990).

[26] **J.C. Picard, M. Queyranne,** *"The Time-Dependent Traveling Salesman Problem and its Application to the Tardiness Problem in One-Machine Scheduling"*, Operations Research **26 (1)**, 86-110 (1978).

[27] **H.N. Psaraftis, J.N. Tsitsiklis,** *"Dynamic Shortest Paths in Acyclic Networks with Markovian Arc Costs"*, Operations Research **41 (1)**, 91-101 (1993).

[28] **J.M. Rousseau, and S. Roy,** *"RAO Répartition Assistée par Ordinateur : la Déscription du Prototype"*, publication CRT-564, Centre de recherche sur les transports, Université de Montréal, Canada, 1988.

[29] **Y. Shen and J.Y. Potvin,** *"A Computer Assistant for Vehicle Dispatching with Learning Capabilities"*, Operations Reaserch **61**, 189-211 (1995).

[30] **M.M. Solomon,** *"Algorithms for the Vehicle Routing and Scheduling Problems with Time Window Constraints"*, Operations Research **35**, 254-265 (1987).

[31] **É. Taillard, P. Badeau, M. Gendreau, F. Guertin and J. Y. Potvin,** *"A Tabu Search Heuristic for the Vehicle Routing Problem with Soft Time Windows"*, Transportation Science **31**, 170-186 (1997).

[32] **A.K. Ziliaskopoulos,** *"Optimum Path Algorithms on Multidimensional Networks : Analysis, Design, Implementation and Computational Experience"*, Ph.D. Dissertation, University of Texas at Austin, 1994.

[33] **A.K. Ziliaskopoulos and H.S. Mahmassani,** *"Time-Dependent, Shortest-Path Algorithm for Real-Time Intelligent Vehicle Highway System Applications"*, Transportation Research Record **1408**, 94-100 (1993).

Chapitre 5

Exploiting Knowledge about Future Demands for Real-Time Vehicle Dispatching

La référence de cet article est :

S. Ichoua, M. Gendreau and J.Y. Potvin, "Exploiting Knowledge about Future Demands for Real-Time Vehicle Dispatching", soumis à Transportation Science en Août 2001.

Exploiting Knowledge about Future Demands for Real-Time Vehicle Dispatching

Soumia Ichoua[1,2], Michel Gendreau[1,2]
and Jean-Yves Potvin[1,2]

1 Centre de recherche sur les transports. Université de Montréal, C.P. 6128, succ. Centre-ville, Montréal, Québec, Canada H3C 3J7.

2 Département d'informatique et de recherche opérationnelle, Université de Montréal, C.P. 6128, succ. Centre-ville, Montréal, Québec, Canada H3C 3J7.

Abstract. An important, but seldom investigated, issue in the field of dynamic vehicle routing and dispatching is how to exploit information about future events to improve decision-making. In this paper, we address this issue in a real-time setting with a strategy based on probabilistic knowledge about demand arrivals to better manage the fleet of vehicles. More precisely, the new strategy introduces dummy customers (representing forecasted requests) in vehicle routes to insure a good coverage of the territory. This strategy is assessed through a set of computational experiments performed in a simulated environment.

5.1 Introduction

The rapid growth in communication and information technologies, as well as the increased interest in just-in-time distribution systems, have recently led researchers to focus on dynamic vehicle routing and scheduling problems. In the static version of these problems, the goal is to design a set of minimum cost routes to satisfy the demand while satisfying various constraints. In a real-time setting, a number of service requests arise continuously and randomly over time. Thus, each new request must be inserted in real-time in the current set of planned routes (where a planned route is a sequence of requests that have been assigned to a vehicle, but have not been serviced yet). Clearly, any solution is based on incomplete information, since some requests have not arisen yet. Furthermore, some of the request attributes, like actual demand or time to start service, may be prone to random variations over time. In this context, uncertainty is an inherent characteristic of the problem.

Fortunately, future events often arise in a somewhat predictable way (e.g., historical averages and probability distributions can be derived from previously collected data). An important issue that has been rarely investigated in the field of dynamic vehicle routing and dispatching is how to exploit probabilistic knowledge about future events to improve the decision-making process. Actually, exploiting information about future events is likely to result in a better management of the resources at the current time, which implies that future needs will be met more efficiently. In this paper, we propose a strategy to exploit probabilistic information about future demand arrivals. The effectiveness of this strategy is assessed within a parallel tabu search heuristic, previously reported in Gendreau et al. (99). This development is motivated from the local operations of long-distance express mail services (e.g., Federal Express). In this application, customers call a central office

when they want some mail to be delivered abroad. Local collection routes are designed to address these customers' requests. The collected mail is then brought back to the central office for further processing and shipping.

The static version of this problem can be stated as follows. A fixed size fleet of m identical vehicles is used to service customer requests. The goal is to design a set of minimum cost routes which service every customer, such that :

- each vehicle route originates from and terminates at a central depot;
- each vehicle services one route;
- each request is served once by exactly one vehicle;

Furthermore, each service point i (including the depot) has its own time window $[e_i, l_i]$, where e_i is the earliest service time and l_i is the latest service time. These time windows are "soft", i.e., a vehicle can arrive before the lower bound or after the upper bound. If the vehicle is too early, it must wait to start its service; if the vehicle is too late, a penalty for lateness is added to the route cost (which is usually related to the total distance traveled by the vehicle). However, the upper bound of the time window at the depot is hard. Thus, a vehicle is not allowed to come back late.

The paper is organized as follows. Section 2 provides a brief literature review dedicated to stochastic vehicle routing problems in static and dynamic settings. In this section, we mostly focus on the exploitation of knowledge about future demands. Section 3 introduces a new probabilistic strategy for managing the fleet of vehicles and discusses different issues related to it. Section 4 briefly describes the original tabu search heuristic reported in Gendreau et al. (99) and explains how it was modified to integrate the new strategy. Section 5 reports computational results obtained with the new algorithm. Finally, section 6 summarizes the major

findings of the paper.

5.2 Literature review

In the field of vehicle routing, the issue of exploiting knowledge about future demands to improve decision-making has gained an increasing interest in the last decade. This issue has been addressed in the literature under two different problem classes (which are associated with two different ways of tackling the uncertainty, one static, the other dynamic). The existing work for each class is reported in the two following subsections.

5.2.1 Static stochastic vehicle routing problems

In this problem class, the customers and/or their demand are discrete or continuous random variables. Probabilistic information about future events is used to construct an "a priori" solution which optimizes the expected value of a given objective function. The probabilistic traveling salesman problem (PTSP), the vehicle routing problem with stochastic demands (VRPSD), the vehicle routing problem with stochastic customers (VRPSC), and the vehicle routing problem with stochastic customers and demands (VRPSCD) all belong to this class. A survey on these problems can be found in Gendreau et al.(96). In the following, these problems are briefly described and an overview of some related work is provided.

The PTSP is defined on a graph $G = (N, A)$, where N is the set of nodes (customers) and A is the set of arcs. A cost c_{ij} is associated with each arc $(i, j) \in A$ and each customer i has a probability p_i of requiring service. The goal is to find a single tour of minimum expected cost. This problem was first introduced by Jaillet (85, 88) and investigated by many researchers (e.g., Jezequel (85), Bertsimas and Howell (93) and Laporte et al. (94)). In the VRPSD, vehicles of fixed capacity must

service customers with a stochastic demand. Some recent papers have been devoted to this problem (Bertsimas (88), Laporte et al. (89), Séguin (94), Gendreau et al. (95) and Yang et al. (2000)). In the VRPSC, each customer i has some probability p_i of requiring service, but the demand is deterministic (Bertsimas (88) and Waters (89)). The VRPSCD combines characteristics of the VRPSC and the VRPSD, given that both the customers and their demand are stochastic. This problem was investigated in Bertsimas et al. (90), Bertsimas (92), Gendreau et al. (94, 95) and Séguin (94).

In these problems, it is assumed that the probability distributions are known and are used to construct a solution in two stages. In a first stage, a planned (or "a priori") route is constructed which minimizes an objective function related to some expected measure (e.g., expected length of the routes). When the routes are actually executed in a second stage, recourse actions are applied to the first stage solution when needed to respond to the current situation (e.g., "jumping over" a customer who does not show up). The interested reader will find more details about this kind of solution approach in Dror et al. (89) and Gendreau et al. (96).

5.2.2 Dynamic stochastic vehicle routing problems

Here, new events are handled as they are revealed over time. Thus, no solution is constructed "a priori" in this case. Assuming that some probabilistic knowledge about future events is available, two different ways of exploiting this information are reported in the literature. This is the subject of the following subsections.

5.2.2.1 Analytical studies

Here, the only source of uncertainty considered is the occurrence of new requests and/or the service (or dwell) times at customers. The probability distri-

bution of these random variables is used to perform analytical studies aimed at examining the robustness and asymptotic behavior of different deterministic and myopic dispatching strategies. Bertsimas and Simchi-Levi (96) pointed out that this kind of work also provides new insights into the solution structure, thereby offering valuable guidelines for the design of new efficient deterministic algorithms. An example is found in Bertsimas and Van Ryzin (91), where the authors consider the dynamic traveling repairman problem (DTRP). In this problem, a vehicle is used to service a set of independently and uniformly distributed customer demands which occur according to a Poisson process with intensity parameter λ. The service time at each customer is also a random variable. The goal is to find a routing policy which minimizes the expected system time, which is the time spent in the system by each customer. Using results from geometrical probability, combinatorial optimization and queuing theory, the authors provide an analytical study of the problem in the case of light traffic and heavy traffic. First, the authors derive lower bounds for the average system time. Then they analyze several simple policies (e.g., First Come First Served) and compare their performance with these lower bounds. In 1993, the same authors (Bertsimas and Van Ryzin (93)) extend their findings to the case of a homogeneous fleet of m vehicles where each vehicle can visit at most q customers before returning to the depot. The uncapacitated version of this problem was also investigated. The authors partition the given area into subregions and use the results of the first paper to propose and analyze new policies in case of light and heavy traffic, respectively.

Swihart and Papastavrou (99) introduce a new variant of the DTRP where each service request has a pick-up and a delivery location. The objective is to minimize the expected system time. The authors consider the unit-capacity case where the

vehicle can carry no more than one item, as well as the case where the vehicle can carry an arbitrary large number of items. In each of these two cases, lower bounds are derived for the average system time in light and heavy traffic conditions. Several policies are also proposed and compared.

In his dissertation, Rubio Maqueo (1995) has studied the vehicle routing and inventory problem (VRIP). In this problem, a single capacitated vehicle operates out of a single warehouse and services a finite set of retailers. The arrival of demands at a given retailer is modeled as an independent renewal process while the quantity of demand is stochastic. Also, the travel times between retailers are random variables. The objective is to minimize the expected inventory and transportation costs. Assuming that the system operates in heavy traffic, the author analyzes two models. In the first one, the vehicle follows a predefined routing scheme : deliver full loads to either a single retailer (direct shipping (DS)), or along a pre-specified TSP tour. In the second model, which is based on the current inventory levels, the controller dynamically chooses whether the vehicle should follow a TSP or a DS routing scheme. Using simulations and queuing theory, the author fully characterizes a dynamic control policy which is asymptotically optimal for the fixed routing scheme. In addition, the author investigates the importance of various operational decisions (e.g., TSP vs. DS, fixed vs. dynamic routing), thereby giving insights into the nature of the optimal solution to the VRIP.

In the first part of his dissertation, Kleywegt (96) introduces and studies a problem called the dynamic and stochastic knapsack problem (DSKP). In this problem, demands for a given resource occur according to some stochastic process. Each demand requires a random amount of resource and has a random profit. Demands can be either accepted or rejected. If a demand is accepted, an associated reward

is gained. Conversely, if it is rejected, a penalty is incurred. There is also a deadline, possibly infinite, after which demands can no longer be accepted (although the process can be stopped at any time before this deadline). The objective is to determine an optimal policy for accepting the demands and for stopping the process while maximizing the expected reward value. The author considers both the infinite and finite horizon cases. He establishes a number of theoretical characteristics for the optimal accepting rule and for the optimal value function, and uses these characteristics to derive an optimal policy for the problem. In the second part of his thesis, the author formulates and analyzes the dynamic and stochastic distribution problem (DSDP) with a network of terminals and a fleet of m vehicles. In this problem, customers call at random times for the transportation of a set of loads from given origins to given destinations. Each accepted request generates a reward and each rejected request incurs a penalty. There is also a holding cost per unit of time which depends on the number of vehicles, as well as the number of loads waiting at each terminal. The objective is to maximize the expected total value (rewards minus costs). The problem is formulated as a Markov decision process. Equations are derived for the optimal expected value and some fundamental structural characteristics of the solution are established. A two-stage algorithm is also proposed. The first stage is a threshold acceptance rule which is shown to be optimal, and the second stage is an optimal dispatching rule.

5.2.2.2 Stochastic approaches

This line of research has sought to develop explicit stochastic and dynamic approaches which incorporate the current information and probabilities of future events to produce more efficient solutions than those obtained through myopic deterministic strategies. The only work that we are aware of in this category is

that of Powell and his team (Powell (88), Powell (96), Powell et al. (2000)), and a recent research done by Larsen (2000). In the following, we briefly discuss these studies.

The problem considered by Powell and his team is motivated from long-haul truckload trucking applications. In this context, the problem consists in dynamically assigning drivers to loads which arise randomly over time. Each load is characterized by its origin, its destination and its time window. At any time, a vehicle can be either empty or carrying a single load. Furthermore, a demand may be rejected if it cannot be serviced within reasonable time. Powell (88) addresses the issue of relocating vehicles in anticipation of future demands. He first proposes a dynamic network model based on a time-space graph where the nodes correspond to known and forecasted demands in different zones at different time periods. In addition to the arcs which represent moves between regions, a dummy arc from each origin to a super-sink captures the impact of the current time period on the future. The cost on this arc is the expected reward value of having an additional vehicle in the associated region at the beginning of the next time period. This model has some limitations since a forecasted load may be serviced even if it never materializes. To overcome this weakness, the author proposes a stochastic, dynamic model where the demand $D_{ij}(t)$ between nodes i and j at time period t is now a random variable. Hence, the flow $x_{ij}(t)$ of loaded vehicles as well as the flow $y_{ij}(t)$ of empty vehicles moving between i and j at the current time period are random variables. This model is written as a two-stage stochastic program where the objective function combines a deterministic cost that corresponds to the first time period (the present) and the value of the recourse function that represents the expected cost from the next time period to the end of the time horizon (the future).

In Powell (96), the author investigates in the same context a hybrid model for solving a combined assignment and fleet management problem. The model is based on a dynamic-stochastic network with two components. The first one is the assignment network aimed at assigning drivers to loads during the current time period. The second is the forecast network which represents the remainder of the time horizon and includes all forecasted demands, as well as known loads to be picked up in the future. Arcs in this part of the network represent loaded and empty moves or a waiting time from one period to the next. A set of links connecting the two parts of the network represent empty relocation moves in anticipation of future loads. The expected recourse function is represented by a cluster of "recourse links" which capture the expected marginal contribution of available vehicles in a given region during a given time period. An approximation of this function value is added to the arc costs and the resulting transportation problem is then solved with a network simplex algorithm on a rolling horizon basis.

In Powell et al. (2000), a new approach is proposed for solving large-scale stochastic and dynamic problems such as integer multi-commodity flow problems. In these problems, new decisions are made continuously over time in response to new information. The approach is first presented within a general framework. Then, the authors report its application in different contexts, like vehicle dispatching. The proposed approach combines techniques from dynamic programming and multistage stochastic programming. More precisely, it uses forward dynamic programming techniques (Bertsekas and Tsitsiklis (96)) that step forward in time to avoid the evaluation of all possible states. The curse of dimensionality, which is inherent to dynamic programming approaches, is thus alleviated.

The issue of relocating an empty vehicle in anticipation of future demands is

also addressed in Larsen (2000), where probabilistic knowledge about future events is exploited. The problem studied is a dynamic traveling salesman problem where the goal is to find a minimum cost tour through a set of dynamic requests. The requests, each with a soft time window, unfold over time in n subregions according to a Poisson process with rate $\lambda_i, i = 1, ..., n$. Each subregion may have one or more resting locations (idle points) for the vehicles. Moreover, all requests must be serviced even if some vehicles arrive late at the depot. The objective function is a weighted sum of total travel time and total lateness over all requests. Whenever a new service request occurs, the current tour is re-optimized using a simple heuristic based on an exchange procedure. Once the vehicle has finished serving its current customer, the algorithm determines the best resting location for the vehicle, if the time window at its next destination does not allow immediate service (and assuming that the probability of having at least one request in the chosen location is high enough). This is achieved using the following on-line policies : **(i)** the vehicle goes to the nearest resting location, **(ii)** the vehicle goes to the busiest resting location (i.e., in the subregion with the largest λ_i value), **(iii)** the vehicle goes to the resting location which belongs to the subregion with the highest expected number of customers. Simulations on problems with about 40 customers were used to compare the different relocation policies including a "no-rest" policy where the vehicle systematically waits at its current position. Unfortunately, the empirical analysis does not clearly identify a dominant policy.

5.3 A probabilistic strategy for managing the fleet of vehicles

5.3.1 Motivation

The challenge of designing models and methods which accurately reflect real-life conditions is an on-going strive in the field of dynamic vehicle dispatching. An important concern in this line of study is the exploitation of knowledge about future events to improve solution quality. In real-life, skilled dispatchers typically end up with some valuable knowledge about market demand (e.g., "peak" time periods, intense geographic zones, etc...). This knowledge allows them to better manage the current resources in anticipation of forecasted needs. Hence, an interesting issue in the field of dynamic vehicle dispatching, is the development of "well informed" solution approaches which explicitly reproduce the dispatchers' behavior. This has been investigated by Powell (86, 96) in the context of a truckload carrier (see section 2). In this case, the problem is a combined pick up and delivery problem with no consolidation (i.e., at any time, a vehicle is either empty or is carrying a single load) and the horizon typically extends over a number of days.

With regard to the work of Larsen (2000), we recall that when the vehicle finishes the service at its current customer, the two proposed policies choose, among all resting locations (including the current position), where the vehicle should spend its waiting time until the beginning of service at its next customer. In fact, it might be better for a vehicle to systematically wait at its current position rather than anywhere else. In the worst case (i.e., when the forecasted demands do not materialize), waiting at a resting location which is different from the current vehicle location simply incurs an additional traveling distance.

As mentioned by Larsen, the proposed policies can also suggest a bad choice,

namely to move the vehicle in a zone which is surrounded by many other zones with low request arrival rates, and far from regions where new demands are very likely to appear. This is illustrated in figure 1. In this figure, the number in each

2	1	4
3	1	1
3	3	2

FIG. 5.1 – Moving the vehicle far from an "interesting" area.

square represents the arrival rate λ in the corresponding zone. In this case, the two policies would choose the zone with $\lambda = 4$, thereby ignoring the "interesting area" in the bottom. To overcome this shortcoming, one has to take into account the arrival rate in neighboring zones. In the following, we propose a probabilistic approach which addresses this issue in a broader context.

5.3.2 The strategy

The proposed strategy exploits probabilistic knowledge about the arrival of future demands to manage the fleet of vehicles more efficiently. The intent is to provide a good coverage of the territory in anticipation of forecasted demands.

5.3.2.1 Basic assumptions

Let us assume that the distribution area Z is partitioned into J zones $Z_1, Z_2, ..., Z_J$ and that the time horizon is divided into K time intervals $I_1, I_2, ..., I_K$. Moreover, we suppose that it is possible to estimate the probability p_{jk} that at least one demand will occur in zone Z_j during time period I_k. If this probability is high, it

might be worthwhile to force the vehicle to wait in this zone instead of directing it towards its next planned destination (if the latter is in another zone). The intent is to keep the vehicle readily available to service demands that are likely to unfold in its current zone.

To avoid directing the vehicle towards "bad" areas (as illustrated in figure 1), the probability that at least one demand will occur is calculated using a location based analysis. That is, instead of applying directly the probability p_{jk} of having at least one demand in the current zone Z_j, the probability is calculated over a neighborhood of the current vehicle position, which may have a non empty intersection with many zones. In the following we discuss the various elements which are taken into account by our new decision rule.

5.3.2.2 Waiting time

Actually, it can be useless for the vehicle to wait at its current location if the probability of occurrence of a new request in its neighborhood in the near future is not high enough. Moreover, while the vehicle is forced to wait, some new "real" requests in other zones, which appear further along its planned route, will suffer from service delays. To overcome this problem, a vehicle i is kept at its current position only for some amount of time δt_i. Parameter δt_i is very important : if it is too small, there may not be enough time for forecasted demands to materialize. If it is too large, the vehicle may wait for a long time for forecasted demands that will never come true, which may lead to additional lateness on its planned route.

In our implementation, $\delta t_i = min(\Delta, \bar{t}_i - t, (\underline{t}_i - t) + \eta)$. The first component Δ is a fixed parameter which is simply aimed at avoiding too large values. The second component is the difference between the latest feasible departure time $\bar{t}_i$ (which allows the vehicle to return before the upper bound of the time window at the

depot) and the current time t. The third component is the difference between the latest departure time which does not induce any additional lateness on the vehicle route $\underline{t}_i$ and the current time t, plus a given parameter η. This last parameter reflects some tolerance to additional lateness.

5.3.2.3 Probability of occurrence of a new request

In our strategy, a given vehicle i may be forced to wait at its current position for some amount of time δt_i. But, because of the probabilistic nature of the available information, these demands may never materialize. In this case, the vehicle will wait for nothing. Thus, this waiting time is allowed only if the probability for a demand to occur in the vehicle neighborhood during time interval $[t, t+\delta t_i]$, where t is the current time, is greater than or equal to a given threshold. In the following, the probability calculation is explained.

We suppose that the service requests occur within a time-space Poisson process where λ_{jk} is the demand rate in zone Z_j during time period I_k, $1 \leq j \leq J$, $1 \leq k \leq K$. The Poisson distribution has some nice properties that lead to easily derived analytical expressions for describing the system. Let us assume that vehicle i has just finished servicing its current customer in zone Z_j. Then, the neighborhood $\bar{N}_i$ of vehicle i is defined as a square centered at the current vehicle position l_i. In our experiments, $\bar{N}_i$ has the same dimension as a zone, although it does not need to be in general. This is illustrated in figure 2 for a distribution area divided into 9 zones. In this figure, the current position l_i is represented by a large dot.
Let also assume that at instant t, vehicle i is ready to leave its current location l_i in zone Z_j. Then, the probability $p^{\delta t_i}$ that at least one demand will occur during time interval $[t, t+\delta t_i]$ in the neighborhood $\bar{N}_i$ is $p^{\delta t_i} = 1 - p_0^{\delta t_i}$, where $p_0^{\delta t_i}$ is the probability that no new request occurs during time interval $[t, t+\delta t_i]$ in $\bar{N}_i$. If

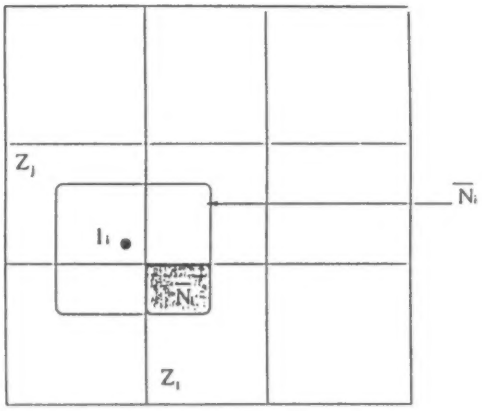

FIG. 5.2 – Defining the vehicle neighborhood $\bar{N}_i$.

we suppose that t belongs to time interval I_{k_1} and that $(t + \delta t_i)$ belongs to time interval I_{k_2} $(k_1, k_2 \in \{1, .., K\})$, then the interval $[t, t + \delta t_i]$ can be partitioned into $(k_2 - k_1 + 1)$ intervals, each of length δt_i^r and belonging to time period I_r $(k_1 \leq r \leq k_2)$. This is illustrated in figure 3 where the time horizon is partitioned into three time periods. In this figure, t belongs to time period I_1 and $(t + \delta t_i)$ belongs to time period I_3.

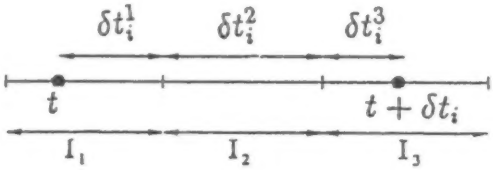

FIG. 5.3 – Partitioning δt_i.

The probability $p_0^{\delta t_i} = \prod_{r=k_1}^{k_2} p_0^{\delta t_i^r}$, where $p_0^{\delta t_i^r} = exp(-\lambda_{\bar{N}_i r} \delta t_i^r)$ is the probability that no new request occurs during δt_i^r in $\bar{N}_i$ and where $\lambda_{\bar{N}_i r}$ is the demand rate in $\bar{N}_i$ during time period I_r. Since the J Poisson processes are assumed to be independent, $\lambda_{\bar{N}_i r} = \sum_{l=1}^{J} \lambda_{lr} \frac{S_{\bar{N}_i}}{S}$.

In this equation, λ_{lr} is the demand rate in zone Z_l during time period I_r ;

$\bar{N}_l = Z_l \cap \bar{N}_i$, $S_{\bar{N}_l}$ is the area of $\bar{N}_l$, and S is the area of a zone.

5.3.2.4 Coverage of the territory

While applying the proposed strategy, it may well happen that most or even all vehicles will be kept in a zone where the demand rate is high. Thereby, a new request which appears in another zone may be serviced very late or not serviced at all. To overcome this shortcoming, the number of vehicles in each zone must always be less than or equal to a given upper bound.

5.3.2.5 Summary of the strategy

Figure 4 gives a pseudo-code description of the proposed strategy for a given vehicle.

1. if $(d \leq \alpha\ \bar{d})$ go to the next customer location (and exit the procedure).
2. if $((prob[$at least one demand in neighborhood during $\delta t] > s)$ & $(v \leq V))$
   wait at the current location for δt time units,
   else go to the next customer location.

FIG. 5.4 – Pseudo-code of the proposed strategy.

In this figure, $\bar{d}$ is the average distance between two consecutive customers locations and d is the distance between the current position of the vehicle and the location of its next destination. Also, v is the number of vehicles in the current zone and V is the upper bound on the number of vehicles in this zone. Finally, α and s are user-defined parameters. Thus, a vehicle waits at its current location if its next destination is far enough, the probability of occurrence of a new request in the vehicle's neighborhood is high enough and there are not too much vehicles in the current zone.

The effectiveness of this rule is assessed by incorporating it within a parallel tabu search heuristic, previously reported in Gendreau et al. (99). The next section will briefly describe the original algorithm and explain how it was modified to integrate the proposed strategy.

5.4 A parallel tabu search algorithm

5.4.1 The original algorithm

Given the dynamic context, a solution at any instant t is a set of planned routes, each beginning with the current destination of the associated vehicle. This current destination is fixed in the procedure to be described and cannot be modified (i.e., the vehicle must reach it). The algorithm proposed in Gendreau et al. (99) is a parallel tabu search heuristic with adaptive memory (Rochat and Taillard (95)). Tabu search is an iterative local search technique where a neighborhood is generated around the current solution and the best solution in this neighborhood becomes the new current solution (even if it does not provide any improvement). By allowing a degradation of the objective, it is possible to escape from bad local optima, as opposed to pure descent methods. The interested reader will find more details about this approach in Glover and Laguna (97).

The algorithm used in Gendreau et al.(99) can be summarized as follows :

- Construct I different initial solutions with a stochastic insertion heuristic (where the choice of the next customer to be inserted is randomized).
- Apply tabu search to each solution and store the resulting routes in the adaptive memory.
- While a stopping criterion is not met do :

- Use the routes stored in the adaptive memory to construct an initial solution.
- Decompose the problem into subproblems obtained through a geographical decomposition procedure.
- Apply tabu search to each subproblem.
- Add the routes of the resulting solution in the adaptive memory.

The objective is to minimize a weighted sum of total distance traveled and total lateness over all customers.

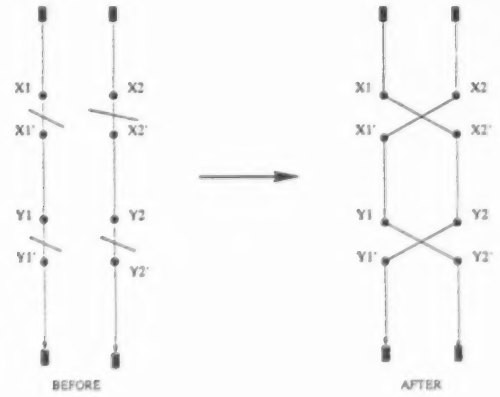

FIG. 5.5 – A CROSS exchange.

The procedure for generating the neighborhood of the current solution is called the CROSS exchange (Taillard et al. (97)). Basically, two segments of variable lengths are taken from two different routes and are swapped (see figure 5).

To speed up the algorithm, a parallel implementation on a network of workstations was developed. The parallelization of the procedure was achieved at two levels :

(1) Different tabu search threads run in parallel, each of them starting from a different initial solution.

(2) Within each search thread, many tabu searches run independently on the

subproblems obtained through the decomposition procedure.

Whenever a new event occurs, the tabu search threads are interrupted. Then, after an appropriate update of the adaptive memory to reflect the current state of the system, the search is restarted with new solutions constructed from the updated memory. Thus, the tabu search is used to improve the current (static) solution between the occurrence of new events, where a new event may be either the occurrence of a new service request or the completion of service at a given customer (since a next destination must be indicated to the driver).

During the schedule generation, a least commitment strategy is considered. That is, if there is some time flexibility with regard to the lower bound of the time window at the next destination, the vehicle waits at its current location (rather than at its next destination).

5.4.2 Implementing the new strategy

The new strategy described in section 3.2.5 basically modifies the way of handling new events. This is explained in the following.

(a) When vehicle i has finished serving a customer, the new strategy is used to decide whether to wait or not. In the latter case, the vehicle is directed toward its next destination as in the original algorithm. In the former case, a dummy customer $D_{\delta t_i}$ is created at the current vehicle position. The intent is to force the vehicle to wait at its current location during time interval $[t, t + \delta t_i]$, where t is the current time and δt_i is calculated as in section 3.2.2, using the best solution in adaptive memory. Then, the dummy customer $D_{\delta t_i}$ is inserted at the beginning of the planned route of vehicle i in every solution in the adaptive memory and solutions that become infeasible are discarded (because of the hard time window at the depot, it may not be possible to wait for δt_i time units at the current vehicle

position). Finally, the search threads are restarted with new solutions constructed from the updated memory and the optimization is performed with the assumption that vehicle i will restart at time $t + \delta t_i$.

(b) When a new request occurs, we first look if it is found in the neighborhood of some waiting vehicles. Among all such vehicles, only those which can start service as soon as they reach the new request are kept. Then, we choose vehicle i for which the least additional cost is incurred in its current route. The new request then replaces the dummy customer $D_{\delta t_i}$. If no vehicle qualifies for the new request, the latter is handled as in the original algorithm. Note that the new strategy offers additional insertion points for a new incoming request. These are located between the current vehicle location and its next destination, for each waiting vehicle.

If vehicle i was not used to service any incoming request when time $(t + \delta t_i)$ is reached (either because no new request occurred in its neighborhood or because it was never chosen), the dummy customer $D_{\delta t_i}$ is removed from its route and the vehicle is sent to its next planned destination, as indicated by the best solution in the adaptive memory.

5.5 Computational results

5.5.1 Simulation framework

Simulations were performed to assess the effectiveness of the proposed strategy. Hence, a simulator was developed to produce different operating scenarios which reflect as closely as possible what is observed in the real world. In the following, we describe the components of this simulator.

5.5.1.1 Time horizon

The simulator first sets up a time horizon which represents the entire labor day. For example, a labor day running from 9h00 to 16h00 is represented by a time horizon of 300 minutes. Within this horizon, the simulator produces discrete time events which correspond to either a request arrival or the end of service at a given customer location. The new request arrivals take place only in the first 240 minutes of the time horizon, to allow vehicles to service their planned routes before returning to the depot.

5.5.1.2 Generation of new service requests

Typically, the arrival rate of new requests varies over time (e.g., peak hours) and space (e.g., densely populated areas). Hence, a time-space Poisson process is used. We recall that the distribution area is partitioned into J zones $Z_1, Z_2, ..., Z_J$ and that the time horizon is divided into K time intervals $I_1, I_2, ..., I_K$. Moreover, λ_{jk} denotes the demand rate in zone Z_j during time period I_k, $1 \leq j \leq J$, $1 \leq k \leq K$. The characteristics of a new service request are generated as follows. Given the current time interval I_k, the demand rate $\lambda_k = \sum_{j=1}^{J} \lambda_{jk}$ of the global distribution area Z (which is made of zones $Z_j, j = 1, 2, ..., J$) over time interval I_k is used to determine the arrival time of the next request. Then, the zone Z_j where the new request will appear is determined using the following cumulative probabilities $(P_j^k)_{0 \leq j \leq J}$:

$$P_j^k = \begin{cases} 0, & \text{if } j = 0 \\ \sum_{l=0}^{j} p_{lk} & \text{otherwise} \end{cases}$$

where $p_{lk} = \frac{\lambda_{lk}}{\sum_{j=1}^{J} \lambda_{jk}}$.

The selected zone Z_j is such that $P_{j-1}^k \leq u < P_j^k$, where u is drawn from a uniform distribution $U[0,1]$. To generate the time window associated with the new request r, the earliest pick-up time e_r is first generated in the interval $[t, t + \Delta T_1]$, where t is the current time and ΔT_1 is a constant. Then, e_r is possibly adjusted to avoid exceeding $\tilde{t} = l_0 - tr(r,0) - s_r$, where l_0 is the end of the day, $tr(r,0)$ is the travel time from r to the depot and s_r is the service or dwell time. Clearly, it is not possible to return to the depot on time if the service at request r starts after $\tilde{t}$. Once the earliest pick-up time e_r is determined, the latest pick-up time l_r is set to $e_r + \Delta T_2$, where ΔT_2 is a constant. The value l_r is adjusted similarly to e_r if necessary.

5.5.2 Testing scenarios

Different sets of problems were generated to perform numerical tests. In these problems, the fleet size is set to 4 or 6 vehicles with all vehicles moving at a constant average speed of 30 kms/hour. The service time is equal to 5 minutes at each customer. Parameters ΔT_1 and ΔT_2 are both set to 30 minutes. The time horizon is set to 300 minutes (i.e., 5 hours) and is divided into three time periods : morning, lunch time and afternoon. The lunch time period is one hour long and the two other ones are two hours long. The distribution area is a 5 km X 5 km square area and the depot is located at (3.3 km, 4.5 km). Nine "active" zones are considered in the distribution area and are partitioned into two categories : category 1 contains the intense central zone where the demand rate is high, while category 2 contains the remaining peripheral zones where the demand rate is low. Moreover, in each problem the set of requests is divided into two subsets. The

first subset contains requests that are known at the beginning of the day (i.e., 25% of the total number of requests). The second subset contains requests that unfold in real-time. In this case, the arrival time of each request is calculated based on a 3x2 time-dependent demand rate matrix $(\lambda_{cT})_{1\leq c\leq 2, 1\leq T\leq 3}$, where λ_{cT} is the demand rate in a zone of category c during time period T. The entries are adjusted to create two different types of scenarios : scenario 1 where the total number of requests is about 180, and scenario 2 where the total number of requests is about 240. In the latter case, a higher degree of dynamism is obtained because there are more real-time requests over the same time horizon (in average, we obtain 27 requests per hour for scenario 1, and 36 requests per hour for scenario 2). The demand rate matrices for the two scenarios, in number of requests per hour, are reported in tables 1 and 2, respectively. It is worth noting that in real-life, new requests rarely unfold in the intense zone (associated with downtown) during the lunch time period. Thus, the corresponding demand rate is set to 0 for the two scenarios during that time period. Tests are performed on 10 instances for each scenario.

	T_1	T_2	T_3
Intense zone	16.50	0.000	33.00
peripheral zones	2.04	2.04	2.04

TAB. 5.1 – *Demand rate matrix in scenario 1.*

	T_1	T_2	T_3
Intense zone	19.98	0.000	39.96
peripheral zones	2.52	4.98	2.52

TAB. 5.2 – *Demand rate matrix in scenario 2.*

5.5.3 Experiments

The experiments reported in this section were performed on a network of 3 SUN UltraSparc-IIi workstations (300 Hz). Each process is programmed in C++ and communication between the processes is handled by the Parallel Virtual Machine Software. As mentioned previously, the objective is to minimize the sum of total travel time and total lateness over all customers. The value of the parameters in the original tabu search algorithm are kept as in Gendreau et al. (99). Parameters related to our decision rule are examined in the following.

5.5.3.1 Preliminary tests

To determine the best values for parameters α, s, η, V and Δ used in our waiting strategy (see figure 4), tests were performed with 4 vehicles over a sample of 4 problems. Several values had to be tested for every problem in the sample. Thereby, a significant amount of computation time is required if the size of the sample is too large. On the other hand, this size has to be large enough to get a correct calibration. Since scenarios 1 and 2 represent different degrees of dynamism, the tests were conducted separately for the two scenarios. Table 3 and 4 present a sample of results obtained with different combinations of values using scenario 1 and scenario 2, respectively. We show only the best combinations of values in these tables (i.e., less than 5 unserved customers on average for scenario 1 and less than 23 unserved customers on average for scenario 2). In these tests, parameter V was set to 1 for the peripheral zones, while different values were tested for the intense central zone. The four numbers in each entry in tables 3 and 4 are the average number of unserved customers, total travel time, total lateness and objective value, respectively. Based on the results obtained, the values $\alpha_1 = 0.20$,

$s_1 = 0.80$, $\eta_1 = 200$ *sec.*, $\Delta_1 = 300$ *sec.* and $V_1 = 1$ for scenario 1 and $\alpha_2 = 0.40$, $s_2 = 0.80$, $\eta_2 = 100$ *sec.*, $\Delta_2 = 180$ *sec.* and $V_2 = 1$ for scenario 2 were finally kept for further investigations. In these experiments, we also found that δt_i is about 4.50 minutes for scenario 1 and 2.90 minutes for scenario 2.

5.5.3.2 Numerical results

The final tests were conducted on 10 instances for each scenario with a fleet size of 4 or 6 vehicles.

Scenario 1

Tables 5 and 7 report results obtained with the original algorithm (Gendreau et al. (99)), with 4 and 6 vehicles, respectively, while tables 6 and 8 report the results obtained with the new algorithm. The results show that the new algorithm performs well against the original one. More precisely :

Experiments with 4 *vehicles*

The new strategy leads to an improvement over 8 of the 10 problems tested. In the case of the two other problems, the results obtained with the new algorithm are as good as those obtained with the original one. Note that among the eight instances where an improvement was observed, the original algorithm succeeds in servicing all customers in five cases. In these cases the new strategy also reduces the objective value by 1.1 % to 12.2 %. In the three other instances, the number of unserved customers is reduced. In two cases, this reduction is achieved a the cost of a marginal increase in the objective value. Note also that the objective value is improved by 4.1 % on average over the 10 problems.

Experiments with 6 *vehicles*

Since these problems are easier than those with only 4 vehicles, the original algorithm succeeds in servicing all customers in 9 problems among the 10 tested. On

two problems, the results obtained with the new algorithm and the original one are the same. On the other hand, the new strategy improves the objective value on six problems. The reduction ranges from 1.1 % to 8.0 %. Moreover, when the original algorithm does not succeed in servicing all customers, the new strategy does. Note that the average improvement in the objective is 2.3 % over the 10 problem instances.

Scenario 2

The problems in this scenario are harder than those of scenario 1 due to the higher demand rates. Hence, the original algorithm never succeeds in servicing all customers. Tables 9 and 11 report results obtained with the original algorithm, with 4 and 6 vehicles, respectively, while tables 10 and 12 report the results obtained with the new algorithm. As expected, the new strategy clearly improves the results obtained with the original algorithm.

Experiments with 4 *vehicles*

In three problems, the two algorithms lead to the same number of unserved customers. However, the new strategy reduces the objective value from 3.0 to 7.5%. In two problems, the increase in the objective value is marginal when compared with the reduction in the number of unserved customers (c.f. 0.2 % against 3.0 % and 0.4 % against 6.3 %, respectively). On the other hand, the new algorithm succeeds in reducing both the number of unserved customers and the objective value on five problems. The reduction ranges from 2.0 % to 16.0 % for the objective value, and from 3.4 % to 29.0 % for the number of unserved customers. In these experiments, the average reduction is 4.2 % for the objective value, and 9.0 % for the number of unserved customers.

Experiments with 6 *vehicles*

Again, the new strategy leads to larger improvements when compared to scenario 1. While the two algorithms both succeed in servicing all customers, the new strategy reduces the objective value in nine cases. The reduction ranges from 2.0 % to 11.0 % with an average of 3.0 %.

In conclusion, the experiments show that the proposed waiting strategy is effective even if it is incorporated into a powerful algorithm, and even if it is enforced only a few times over the scheduling horizon, namely, one to four times per problem for scenario 1, and one to six times per problem for scenario 2. This strategy is also more effective when it is applied on harder problems (i.e., smaller fleet size or higher demand rates).

$\alpha_1 = 0.20$ $s_1 = 0.60$ $\eta_1 = 200$ $V_1 = 1$ $\Delta_1 = 300$	$\alpha_1 = 0.20$ $s_1 = 0.80$ $\eta_1 = 100$ $V_1 = 1$ $\Delta_1 = 180$	$\alpha_1 = 0.20$ $s_1 = 0.80$ $\eta_1 = 200$ $V_1 = 1$ $\Delta_1 = 300$	$\alpha_1 = 0.20$ $s_1 = 0.80$ $\eta_1 = 300$ $V_1 = 1$ $\Delta_1 = 600$
4.00 [1]	4.66	**4.00**	4.66
227.813 [2]	224.70	**229.550**	223.936
234.413 [3]	244.67	**179.980**	207.950
462.226 [4]	469.46	**409.530**	431.890

$\alpha_1 = 0.20$ $s_1 = 0.90$ $\eta_1 = 200$ $V_1 = 1$ $\Delta_1 = 300$	$\alpha_1 = 0.20$ $s_1 = 0.90$ $\eta_1 = 300$ $V_1 = 2$ $\Delta_1 = 600$	$\alpha_1 = 0.30$ $s_1 = 0.90$ $\eta_1 = 200$ $V_1 = 2$ $\Delta_1 = 300$	$\alpha_1 = 0.40$ $s_1 = 0.80$ $\eta_1 = 100$ $V_1 = 1$ $\Delta_1 = 180$
4.00 [1]	4.66	4.33	4.66
225.810 [2]	225.353	229.623	235.08
223.780 [3]	271.556	244.913	248.92
449.590 [4]	496.909	474.536	484.01

$\alpha_1 = 0.50$ $s_1 = 0.60$ $\eta_1 = 200$ $V_1 = 1$ $\Delta_1 = 300$	$\alpha_1 = 0.50$ $s_1 = 0.90$ $\eta_1 = 200$ $V_1 = 1$ $\Delta_1 = 300$	$\alpha_1 = 1.00$ $s_1 = 0.80$ $\eta_1 = 200$ $V_1 = 1$ $\Delta_1 = 300$	$\alpha_1 = 1.00$ $s_1 = 0.80$ $\eta_1 = 200$ $V_1 = 2$ $\Delta_1 = 300$	$\alpha_1 = 1.00$ $s_1 = 0.90$ $\eta_1 = 200$ $V_1 = 1$ $\Delta_1 = 300$
4.33 [a]	4.66	4.66	4.33	4.33
229.588 [b]	226.335	230.051	223.710	222.15
204.350 [c]	209.080	173.428	223.000	218.16
433.940 [d]	435.415	403.480	446.710	440.310

[a]Nb. of unserved customers.
[b]Total travel time.
[c]Total lateness.
[d]Objective value.

TAB. 5.3 – *Best parameter values with 4 vehicles for scenario 1*

$\alpha_2 = 0.20$	$\alpha_2 = 0.20$	$\alpha_2 = 0.20$	$\alpha_2 = 0.20$	$\alpha_2 = 0.20$
$s_2 = 0.65$	$s_2 = 0.65$	$s_2 = 0.80$	$s_2 = 0.80$	$s_2 = 0.80$
$\eta_2 = 60$	$\eta_2 = 200$	$\eta_2 = 100$	$\eta_2 = 180$	$\eta_2 = 200$
$V_2 = 1$	$V_2 = 1$	$V_2 = 1$	$V_2 = 1$	$V_2 = 2$
$\Delta_2 = 100$	$\Delta_2 = 300$	$\Delta_2 = 300$	$\Delta_2 = 300$	$\Delta_2 = 300$
21.33 [1]	22.66	22.66	22	22.66
210.30 [2]	211.48	217.03	214.01	213.08
2423.00 [3]	2385.84	2357.01	2344.13	2420.80
2633.20 [4]	2597.34	2574.10	2558.02	2633.89

$\alpha_2 = 0.40$	$\alpha_2 = 0.40$	$\alpha_2 = 0.40$	$\alpha_2 = 0.40$	$\alpha_2 = 0.40$
$s_2 = 0.60$	$s_2 = 0.80$	$s_2 = 0.80$	$s_2 = 0.80$	$s_2 = 0.80$
$\eta_2 = 60$	$\eta_2 = 60$	$\eta_2 = 60$	$\eta_2 = 100$	$\eta_2 = 100$
$V_2 = 1$	$V_2 = 1$	$V_2 = 1$	$V_2 = 1$	$V_2 = 1$
$\Delta_2 = 100$	$\Delta_2 = 100$	$\Delta_2 = 180$	$\Delta_2 = 180$	$\Delta_2 = 300$
21.66 [1]	21.66	21.66	**21**	22
217.00 [2]	212.20	216.72	**217.10**	214.64
2345.50 [3]	2346.00	2489.40	**2345.90**	2371.58
2562.40 [4]	2558.14	2706.01	**2563.13**	2586.22

$\alpha_2 = 0.40$	$\alpha_2 = 0.40$	$\alpha_2 = 0.40$	$\alpha_2 = 0.40$	$\alpha_2 = 0.40$
$s_2 = 0.80$	$s_2 = 0.80$	$s_2 = 0.80$	$s_2 = 0.85$	$s_2 = 0.90$
$\eta_2 = 180$	$\eta_2 = 200$	$\eta_2 = 200$	$\eta_2 = 60$	$\eta_2 = 100$
$V_2 = 1$	$V_2 = 1$	$V_2 = 2$	$V_2 = 1$	$V_2 = 1$
$\Delta_2 = 300$	$\Delta_2 = 300$	$\Delta_2 = 300$	$\Delta_2 = 100$	$\Delta_2 = 180$
22.66 [a]	22.66	22.66	21.33	22
217.30 [b]	215.42	215.24	216.50	213.30
2429.01 [c]	2424.14	2397.37	2357.50	2573.00
2646.10 [d]	2639.57	2612.62	2574.05	2786.30

[a]Nb. of unserved customers.
[b]Total travel time.
[c]Total lateness.
[d]Objective value.

TAB. 5.4 – *Best parameter values with 4 vehicles for scenario 2*

Instance number	*Number of customers*	*Number of unserved customers*	*Travel Time*	*Lateness*	*Objective Value*
1	152	12	243.24	167.67	410.91
2	140	1	244.58	229.06	473.65
3	127	0	200.84	143.21	344.05
4	144	0	240.81	281.25	522.06
5	132	0	230.15	119.24	349.39
6	120	0	207.83	221.58	429.41
7	126	0	233.27	57.97	291.24
8	155	3	243.96	801.50	1045.47
9	153	3	243.99	436.17	680.16
10	132	0	232.95	89.88	322.84
Average	138.10	1.9	232.16	254.75	486.92

TAB. 5.5 – *Original algorithm under scenario 1 : Simulation of 5 hours with 4 vehicles and 27 requests per hour on average.*

Instance number	*Number of customers*	*Number of unserved customers*	*Travel Time*	*Lateness*	*Objective Value*
1	152	11	233.59	196.43	430.09
2	140	1	243.74	230.13	473.87
3	127	0	202.67	137.67	340.35
4	144	0	227.30	247.09	474.40
5	132	0	219.55	120.24	339.69
6	120	0	201.78	231.42	433.20
7	126	0	224.81	52.09	276.91
8	155	2	239.43	753.19	992.62
9	153	1	240.78	459.54	700.20
10	132	0	238.67	44.49	283.17
Average	138.10	1.5	227.23	247.93	474.48

TAB. 5.6 – *New algorithm under scenario 1 : Simulation of 5 hours with 4 vehicles and 27 requests per hour on average.*

Instance number	*Number of customers*	*Number of unserved customers*	*Travel Time*	*Lateness*	*Objective Value*
1	152	3	248.11	25.84	273.98
2	140	0	232.64	85.79	318.43
3	127	0	218.84	90.70	309.50
4	144	0	233.54	93.37	327.91
5	132	0	226.65	29.47	256.13
6	120	0	198.02	107.32	305.34
7	126	0	217.74	29.93	247.67
8	155	0	255.32	52.53	307.85
9	153	0	235.88	40.82	276.71
10	132	0	221.48	21.69	243.18
Average	138.10	0.3	228.83	57.85	286.60

TAB. 5.7 – *Original algorithm under scenario 1 : Simulation of 5 hours with 6 vehicles and 27 requests per hour on average.*

Instance number	*Number of customers*	*Number of unserved customers*	*Travel Time*	*Lateness*	*Objective Value*
1	152	0	240.24	54.57	294.82
2	140	0	236.13	83.99	320.12
3	127	0	219.57	67.48	287.06
4	144	0	224.68	91.90	316.58
5	132	0	208.29	40.21	248.51
6	120	0	198.49	103.53	302.02
7	126	0	199.25	28.88	228.14
8	155	0	243.26	42.44	285.71
9	153	0	223.01	48.26	271.30
10	132	0	220.61	24.84	245.45
Average	138.10	0	221.35	58.61	279.97

TAB. 5.8 – *New algorithm under scenario 1 : Simulation of 5 hours with 6 vehicles and 27 requests per hour on average.*

Instance number	*Number of customers*	*Number of unserved customers*	*Travel Time*	*Lateness*	*Objective Value*
1	168	7	236.80	1258.40	1495.13
2	196	25	231.80	3077.60	3309.30
3	197	36	192.25	3249.91	3442.20
4	177	17	231.50	2158.10	2389.53
5	182	21	207.00	2732.53	2939.50
6	198	32	214.10	3395.54	3609.62
7	195	38	180.34	3284.86	3465.20
8	164	7	223.92	1391.63	1615.55
9	187	29	188.70	3275.00	3464.00
10	193	29	209.65	2374.30	2584.00
Average	187.00	24.10	211.60	2620.00	2831.40

TAB. 5.9 – *Original algorithm under scenario 2 : Simulation of 5 hours with 4 vehicles and 36 requests per hour on average.*

Instance number	*Number of customers*	*Number of unserved customers*	*Travel Time*	*Lateness*	*Objective Value*
1	168	7	241.40	1201.76	1443.24
2	196	21	218.72	2578.63	2797.35
3	197	35	191.40	3257.40	3448.82
4	177	14	206.70	2119.40	2326.11
5	182	21	203.01	2518.18	2721.20
6	198	30	204.55	3419.40	3623.96
7	195	38	181.90	3178.28	3360.19
8	164	6	232.00	1277.83	1509.85
9	187	27	192.85	3206.80	3399.66
10	193	28	205.63	2286.77	2492.41
Average	187.00	22.07	207.82	2504.44	2712.28

TAB. 5.10 – *New algorithm under scenario 2 : Simulation of 5 hours with 4 vehicles and 36 requests per hour on average.*

Instance number	*Number of customers*	*Number of unserved customers*	*Travel Time*	*Lateness*	*Objective Value*
1	168	0	251.00	283.55	534.55
2	196	0	301.10	577.42	878.50
3	197	0	274.93	1194.33	1469.30
4	177	0	262.10	567.40	829.43
5	182	0	270.54	440.02	710.60
6	198	0	265.32	560.43	825.80
7	195	0	295.90	679.10	974.96
8	164	0	256.40	364.43	621.00
9	187	0	273.80	665.00	940.00
10	193	0	246.30	557.01	803.30
Average	187.00	0	270.00	589.01	859.04

TAB. 5.11 – *Original algorithm under scenario 2 : Simulation of 5 hours with 6 vehicles and 36 requests per hour on average.*

Instance number	*Number of customers*	*Number of unserved customers*	*Travel Time*	*Lateness*	*Objective Value*
1	168	0	265.00	259.21	524.02
2	196	0	280.78	570.55	851.33
3	197	0	269.19	1377.41	1646.61
4	177	0	253.80	519.20	773.01
5	182	0	282.27	371.80	654.07
6	198	0	274.25	521.44	795.70
7	195	0	294.88	650.37	945.25
8	164	0	239.67	345.90	585.57
9	187	0	265.76	601.46	867.23
10	193	0	245.12	470.71	715.84
Average	187.00	0	267.07	568.80	835.86

TAB. 5.12 – *New algorithm under scenario 2 : Simulation of 5 hours with 6 vehicles and 36 requests per hour on average.*

5.6 Conclusion

In this paper, a new strategy which exploits probabilistic knowledge about future demand arrivals to better manage the fleet of vehicles is proposed. Some issues related to this strategy were addressed and investigated. Then, experiments were performed to assess its effectiveness within a previously reported tabu search heuristic. The results show that the proposed strategy provides significant improvements over the original algorithm, especially in some "critical" situations (i.e., small fleet size and high demand rates).

Bibliographie

[1] **D. Bertsekas and J. Tsitsiklis,** *"Neuro-Dynamic Programming"*, Athena Scientific, Belmont, MA. (1996).

[2] **D.J. Bertsimas,** *"Probabilistic Combinatorial Optimization Problems"*, Ph.D. Dissertation, Report No. 193, Operations Research Center, Massachusetts Institute of Technology, 1988.

[3] **D.J. Bertsimas,** *"A Vehicle Routing Problem with Stochastic Demand"*, Operations Research **40**, 574-585 (1992).

[4] **D.J. Bertsimas and L.H. Howell,** *"Further Results on the Probabilistic Traveling Salesman Problem"*, Eur. J. Oper. Res. **65**, 68-95 (1993).

[5] **D.J. Bertsimas, P. Jaillet and A.R. Odoni,** *"A Priori Optimization"*, Operations Research **38 (6)**, 1019-1033 (1990).

[6] **D.J. Bertsimas and D. Simchi-Levi,** *"A New Generation of Vehicle Routing Research : Robust Algorithms, Addressing Uncertainty"*, Operations Research **44**, 286-304 (1996).

[7] **D. Bertsimas and G. Van Ryzin,** *A Stochastic and Dynamic Vehicle Routing Problem in the Euclidian Plane*, Operations Research **39**, 601-615 (1991).

[8] **D.J. Bertsimas and G. Van Ryzin,** *Stochastic and Dynamic Vehicle Routing in the Euclidian Plane with Multiple Capacitated vehicles*, Operations Research **41**, 60-76 (1993).

[9] **M. Dror, G. Laporte and P. Trudeau,** *"Vehicle Routing with Stochastic Demands : Properties and Solution Frameworks"*, Transportation Science **23 (3)**, 166-176 (1989).

[10] **M. Gendreau, F. Guertin, J.Y. Potvin and É. Taillard,** *"Parallel Tabu Search for Real-Time Vehicle Routing and Dispatching"*, Transportation Science **33 (4)**, 381-390 (1999).

[11] **M. Gendreau, G. Laporte, and R. Séguin,** *"A Tabu Search Heuristic for the Vehicle Routing Problem with Stochastic Demands and Customers"*, Operations Research **44**, 469-477 (1996).

[12] **M. Gendreau, G. Laporte, and R. Séguin,** *"An exact Algorithm for the Vehicle Routing Problem with Stochastic Customers and Demands"*, Transportation Science **29**, 143-155 (1995).

[13] **M. Gendreau, G. Laporte, and R. Séguin,** *"Stochastic Vehicle Routing"*, Eur. J. Oper. Res. **88**, 3-12 (1996).

[14] **F. Glover and M. Laguna,** *Tabu Search*, Kluwer, 1997.

[15] **P. Jaillet,** *"Probabilistic Traveling Salesman Problems"*, Ph.D. Dissertation, Massachusetts Institute of Technology, 1985.

[16] **P. Jaillet,** *"A Priori Solution of a Traveling Salesman Problem in which a Random Subset of the Customers Are Visited"*, Operations Research **36**, 929-936 (1988).

[17] **A. Jezequel,** *"Probabilistic Vehicle Routing Problems"*, Master's Thesis, Massachusetts Institute of Technology, 1985.

[18] **A.J. Kleywegt,** *"Dynamic and Stochastic Models with Freight Distribution Applications"*, Ph.D. Dissertation, Massachusetts Institute of Technology, 1996.

[19] **G. Laporte, F.V. Louveaux and H. Mercure,** *"A Priori Optimization of the Probabilistic Traveling Salesman Problem"*, Operations Research **42**, 543-549 (1994).

[20] **A. Larsen,** *"The Dynamic Vehicle Routing Problem"*, Ph.D. Dissertation, Technical University of Denmark, 2000.

[21] **W. Powell,** *A Comparative Review of Alternative Algorithms for the Dynamic Vehicle Allocation Problem*, in Vehicle Routing : Methods and Studies, B.L. Golden and A.A. Assad (eds), pp. 249-291, North Holland : Amsterdam (1988).

[22] **W.B. Powell,** *"A Stochastic Formulation of the Dynamic Assignment Problem, with an Application to Truckload Motor Carriers"*, Transportation Science **30 (3)**, 195-219 (1996).

[23] **W.B. Powell, G. Godfrey, K. Papadaki, M. Spivey and H. Topaloglu,** *"Adaptive Dynamic Programming for Multistage Stochastic Resource Allocation"*, Department of Operations Research and Financial Engineering, Princeton University (2000).

[24] **W.B. Powell, P. Jaillet and A. Odoni,** *Stochastic and Dynamic Networks and Routing*, in Network Routing, Handbooks in Operations Research and Management Science, Volume 8, M.O. Ball, T.L. Magnanti, C.L. Monma and G.L. Nemhauser (eds), pp. 141-295, North-Holland : Amsterdam (1995).

[25] **Y. Rochat and E. Taillard,** *"Probabilistic Diversification and intensification in Local Search for Vehicle Routing"*, Journal of Heuristics, **1**, 147-167

(1995).

[26] **R. Rubio Maqueo,** *"Dynamic-Stochastic Vehicle Routing and Inventory Problem"*, Ph.D. Dissertation, Massachusetts Institute of Technology, 1995.

[27] **R. Séguin,** *"Problèmes Stochastiques de Tournées de véhicules"*, Ph.D. Dissertation, Département d'Informatique et de Recherche Opérationnelle, Université de Montréal (1994).

[28] **M.R. Swihart and J.D. Papastavrou,** *"A Stochastic and Dynamic Model for the Single-Vehicle Pick-up and Delivery Problem"*, Eur. J. Oper. Res. **114**, 447-464 (1999).

[29] **É. Taillard, P. Badeau, M. Gendreau, F. Guertin and J. Y. Potvin,** *"A Tabu Search Heuristic for the Vehicle Routing Problem with Soft Time Windows"*, Transportation Science **31**, 170-186 (1997).

[30] **W.H. Yang, K. Mathur and R.H. Ballou,** *"Stochastic Vehicle Routing Problem with Restocking"*, Transportation Science **34 (1)**, 99-112 (2000).

[31] **C.D.J. Waters,** *"Vehicle Routing Problems with Uncertainty and Omitted Customers"*, Journal of the Operational Research Society **40**, 1099-1108 (1989).

Chapitre 6

Conclusion

Cette thèse est consacrée à l'étude des problèmes de gestion de flottes de véhicules en temps réel. Que l'on soit intéressé par le transport de biens ou de personnes, un grand défi à l'heure actuelle est d'élaborer des approches de résolution rapides mais efficaces dans un environnement qui évolue dans le temps. Notre thèse s'oriente vers ce nouvel axe de recherche qui est en plein expansion grâce aux progrès technologiques récents (p.e. positionnement par satellite, télé-détection, etc...). Ainsi, nous nous sommes attardés sur trois problématiques rencontrées dans des applications temps-réel et nous avons proposé des stratégies de résolution.
Dans ce qui suit, nous présentons un résumé des principales contributions de cette thèse et nous suggérons de nouvelles avenues de recherche qu'il serait opportun d'explorer.

6.1 Principales contributions

Dans le chapitre 2, nous avons présenté une revue et une synthèse des réalisations les plus pertinentes dans le domaine de la répartition des véhicules en temps réel. Dans un premier temps, nous avons dressé un inventaire non exhaustif des

contextes d'application les plus répandus dans la littérature. Par la suite, nous avons classifié ces différents groupes de problèmes en nous basant sur une taxinomie établie au préalable. Nous avons également analysé les types d'approches les plus utilisées dans la littérature et nous en avons dégagé quelques-unes qu'il serait pertinent d'approfondir dans le cadre de travaux futurs.

Le chapitre 3 a porté sur l'étude de la diversion. Notre motivation principale était d'explorer la portée de ce concept qui nous semblait bénéfique pour l'élaboration de stratégies efficaces, capables de mieux gérer les nouvelles difficultés qui surgissent dans un environnement dynamique. Dans cet esprit, nous avons présenté une vision plus large du concept de la diversion. Nous avons également examiné différentes façons de l'intégrer à des algorithmes qui ne sont pas initialement conçus pour en tenir compte. Finalement, nous avons évalué les bénéfices de son utilisation.

Dans le chapitre 4, nous nous sommes penchés sur une problématique de modélisation du problème de répartition de véhicules qui est plus fidèle à la réalité. Ainsi, nous avons proposé et analysé un modèle basé sur des temps de parcours variables dans le temps et qui s'applique dans un contexte statique ou dynamique. Nous avons également démontré qu'il est possible d'adapter des approches de résolution existantes afin de gérer cet aspect de "dépendance du temps" inhérent aux problèmes concrets. Finalement, nous avons comparé le modèle proposé à un modèle qui suppose que les temps de parcours sont constants.

Dans le chapitre 5, nous avons proposé une stratégie qui exploite une connaissance probabiliste des demandes futures en vue d'améliorer le processus de prise de décisions. En particulier, la stratégie proposée introduit des demandes fictives (représentant des requêtes prévues) dans les routes des véhicules afin d'assurer

une bonne couverture territoriale. Nous avons également évalué les bénéfices de l'inclusion d'une telle stratégie dans le processus de prise de décisions.

Les différentes problématiques étudiées dans les chapitres 3, 4 et 5 sont abordées en incorporant les stratégies de résolution proposées au sein de méthodes heuristiques basées sur la recherche avec tabous. Des implantations parallèles sont réalisées afin d'accélérer les temps de réponse, ce qui est très important dans un contexte "temps réel". Enfin, des tests numériques sont réalisés à l'aide de simulations qui s'inspirent de situations réelles observées dans les systèmes de courrier rapide. Les résultats numériques confirment la pertinence des différentes stratégies proposées.

6.2 Nouvelles avenues de recherche

Le but principal de notre recherche était de contribuer à l'élaboration d'approches de résolution qui reflètent la réalité de façon plus adéquate. Ainsi, nous avons modélisé quelques unes des problématiques importantes observées dans des applications concrètes et nous avons proposé des stratégies pour les résoudre. Suite à ce travail, plusieurs avenues de recherche s'ouvrent vers l'élaboration d'un système d'aide à la décision encore plus complet et plus conforme à la réalité. Parmi ces nouveaux développements, mentionnons :

- L'établissement d'une taxinomie plus complète qui permettrait de mieux caractériser les différents types de problèmes de répartition de véhicules en temps réel. Une telle taxinomie serait un pas remarquable vers le développement d'approches de résolution adaptées à chacune des classes de problèmes identifiées.
- La prise en compte d'autres aspects stochastiques du problème comme le

bris d'un véhicule, des accidents de la route, la congestion, les conditions météorologiques, des clients qui annulent leur demande subitement, etc... Ces événements ont des effets aussi bien sur la qualité de service (p.e. retard dans la livraison ou la prise d'une demande, ou parfois son annulation) que sur les coûts d'opérations. Le défi consiste à identifier des stratégies efficaces capables de minimiser ces effets dans un temps raisonnable. À cette fin, il serait judicieux d'établir dans un premier temps une taxinomie de ces événements qui aidera à déterminer les stratégies appropriées pour chacun des types d'événements identifiés.

- L'extension des travaux rapportés dans cette thèse pour des problèmes avec requêtes à deux points. En effet, une telle extension présente de nouveaux défis. Par exemple, dans le cas de la diversion, il faudrait sans doute distinguer les cas où la destination planifiée est un point de collecte ou un point de livraison. Par ailleurs, la définition du délai accordé à l'évaluation de la diversion doit tenir compte des contraintes temporelles associées aux deux points de service. Dans le cas de l'exploitation d'une connaissance probabiliste sur les demandes futures, une re-définition du voisinage de la requête courante s'impose. D'autre part, tout comme pour la diversion, le temps d'attente à la position courante doit être lié aux contraintes temporelles associées aux deux points de service.

Bibliographie

[1] **P. Badeau, M. Gendreau, F. Guertin, J.Y. Potvin, E. Taillard,** *A Parallel Tabu Search Heuristic for the Vehicule Routing Problem with Time Windows*, Transportation Research **5c** (1997), 109-122.

[2] **M.O.Ball, T.L. Magnanti, C.L. Monna and G.L. Nemhauser (eds),** *Network Routing*, Handbooks in Operations Research and Management Science, Volume 8, North-Holland : Amsterdam (1995).

[3] **D.O. Baush, G. G. Brown, D. Ronen,** *Consolidating and Dispatching Truck Shipment of Mobil Heavy Petroeum Products*, Interfaces **25** (1995), 01-17.

[4] **W.J. Bell, L.M. Dalberto, M.L. Fisher, A.J. Greenfield, R. Jaikumar, P. Kedia, R.G. Macj, P.J. Prutzman,** *Improving the Distribution of Industrial Gases with an On-Line Computrized Routing and Scheduling Optimizer* , Interfaces **13** (1983), 04-23.

[5] **R.E. Bellman,** *On a Routing Problem* , Quarterly of Applied Mathematics **16** (1958), 87-90.

[6] **I. Benyahia and J.Y. Potvin,** *Decision Support for Vehicule Dispatching Using Genetic Programming* , IEEE Transactions on Systems, Man and Cybernatics **28** (1998), 306-314.

[7] **D. Bertsekas and J. Tsitsiklis,** *"Neuro-Dynamic Programming"*, Athena Scientific, Belmont, MA. (1996).

[8] **D.J. Bertsimas,** *"Probabilistic Combinatorial Optimization Problems"*, Ph.D. Dissertation, Report No. 193, Operations Research Center, Massachusetts Institute of Technology, 1988.

[9] **D.J. Bertsimas,** *"A Vehicle Routing Problem with Stochastic Demand"*, Operations Research **40**, 574-585 (1992).

[10] **D.J. Bertsimas and L.H. Howell,** *"Further Results on the Probabilistic Traveling Salesman Problem"*, Eur. J. Oper. Res. **65**, 68-95 (1993).

[11] **D.J. Bertsimas, P. Jaillet and A.R. Odoni,** *"A Priori Optimization"*, Operations Research **38 (6)**, 1019-1033 (1990).

[12] **D. Bertsimas and G. Van Ryzin,** *A Stochastic and Dynamic Vehicule Routing Problem in the Euclidian Plane*, Operations Research **39** (1991), 601-615.

[13] **D.J. Bertsimas and G. Van Ryzin,** *Stochastic and Dynamic Vehicule Routing in the Euclidian Plane with Multiple Capacitated vehicules*, Operations Research **41** (1993), 60-76.

[14] **D.J. Bertsimas and D. Simchi-Levi,** *A New Generation of Vehicle Routing Research : Robust Algorithms, Addressing Uncertainty*, Operations Research **44** (1996), 286-304.

[15] **D. Bertsekas and J. Tsitsiklis,** *"Neuro-Dynamic Programming"*, Athena Scientific, Belmont, MA. (1996).

[16] **J.R. Birge and J.K. Ho,** *Optimal Flows in Stochastic Dynamic Networks with congestion*, Operations Research, **41** (1993), 203-216.

[17] **J.Y. Blais, J.M. Rousseau,** *HASTUS : A Model for the Economic Evaluation of Drivers' Collective Agreements in Transit Companies*, INFOR **20** (1982), 03-15.

[18] **L. Bodin, B.L. Golden, A. Assad and M.O. Ball,** *Routing and Scheduling of Vehicules and Crews : the State of the Art*, Computers and Operations Research **10** (1983), 62-212.

[19] **J.W. Braklow, W.W. Graham, S.M. Hassler, K.E. Peck and W.B. Powell,** *Interactive Optimization Improves Service and Performance for Yellow Freight System*, Interfaces **22** (1992), 147-172.

[20] **L. Brotcorne, L. Farand, G. Laporte and F. Semet,** *Impacts des Nouvelles Technologies sur la Gestion des Systèmes de Véhicules d'Urgence*, Technical Report CRT-99-39, Centre de recherche sur les transports, Université de Montréal, Canada (1999).

[21] **G.G. Brown, C. J. Ellis, G. Lenn, W. Graves, David Ronen,** *Real Time, Wide Area Dispatch of Mobil Tank Trucks* , Interfaces **17** (1987), 107-120.

[22] **I. Chabini,** *"Fastest Routes in Temporal Networks Revisited"*, Presented at *Optimization days*, Montreal, May 1966.

[23] **I. Chabini,** *"A New Algorithm for Shortest Paths in Discrete Dynamic Networks"*, Proc. 8th IFAC Symposium on Transportation Systems, Chania, Greece, 551-556 (1997).

[24] **K.L. Cooke and E. Halsey,** *"The Shortest Route Through a Network with Time-Dependent Internodal Transit Times"*, Journal of Math. Anal. Appl. **14**, 492-498 (1966).

[25] **C.F. Daganzo,** *An Approximate Analytic Model of Many-To-Many Demand Responsive Transportation Systems* , Transportation Reaserch **12** (1978), 325-333.

[26] **A. De Palma, P. Hansen, M. Labbé,** *"Commuters' Paths with Penalties for Early or Late Arrival Time"*, Transportation Science **24 (4)**, 276-286 (1990).

[27] **M. Desrochers, J. Desrosiers and M. Solomon,** *A New Optimization Algorithm for the Vehicule Routing Problem with Time Windows*, Operations Reaserch **40** (1992), 342-354.

[28] **J. Desrosiers, Y. Dumas, M.M. Solomon and F. Soumis,** *Time Constrained Routing and Scheduling*, in Handbooks in Operations Research and Management Science, Volume 8 : Network Routing, M.O. Ball, T.L. Magnanti, C.L. Monma and G.L. Nemhauser (eds), pp. 35-139, North-Holland : Amsterdam (1995).

[29] **Robert B. Dial,** *Autonomious Dial-A-Ride Transit Intoductory Overview*, Transportation Reaserch **03C** (1995), 261-275.

[30] **M. Dror, G. Laporte and P. Trudeau,** *"Vehicle Routing with Stochastic Demands : Properties and Solution Frameworks"*, Transportation Science **23 (3)**, 166-176 (1989).

[31] **M. Dror and W. Powell,** *Special Issue in Stochastic and Dynamic Models in Transportation* ,Operations Research **41** (1993), 11-17.

[32] **Y. Dumas,** *Confection d'Itinéraires pour le Transport Adapté*, Publication G-88-44, GERAD, H.E.C., Canada (1988).

[33] **S.E. Dreyfus,** *"An Appraisal of Some Shortest Path Algorithms"*, Operations Research **17**, 395-412 (1969).

[34] **M.L. Fisher, A. J. Greenfield, R. Jaikumar and P. Kedia,** *Real Time Scheduling of a Bulk-Delivery Fleet : Practical Application of Lagrangian Relaxation*, Report 82-10-11, Decision Sciences Dept., University of Pennsylvania (1982).

[35] **M.L. Fisher, H.J. Gang and T.B. Xing,** *Scheduling Bulk-Pickup-Delivery Vehicules in Shangai* , Interfaces **16** (1986), 18-23.

[36] **L.R. Ford Jr. and D.R. Fulkerson,** *"Constructing Maximal Dynamic Flows from Static Flows"*, Operations Research **6**, 419-433 (1958).

[37] **K.R. Fox, B. Gavish, S.C. Graves,** *"An n-Constraint Formulation of the (Time-Dependent) Traveling Salesman Problem"*, Operations Research **28 (4)**, 1018-1021 (1980).

[38] **M. Gendreau, P. Badeau, F. Guertin, J.Y. Potvin, É. Taillard,** *A Solution Procedure for Real-time Routing and Dispatching of Commercial Vehicules*, Technical Report CRT-96-24, Centre de recherche sur les transports, Université de Montréal, Canada (1996).

[39] **M. Gendreau, A. Hertz, G. Laporte and M. Stan,** *A Generalized Insertion Heuristic for the Traveling Salesman Problem with Time Windows*, Operations Research **43** (1998), 330-335.

[40] **M. Gendreau, F. Guertin, J.Y. Potvin and R. Séguin,** *Neighborhood Serach Heuristics for a Dynamic Vehicle Dispatching Problem with Pick-Ups and Deliveries*, Technical Report CRT-98-10, Centre de recherche sur les transports, Université de Montréal, Canada (1998).

[41] **M. Gendreau, F. Guertin, J.Y. Potvin and É. Taillard,** *Parallel Tabu Search for Real-Time Vehicle Routing and Dispatching*, Transportation Science **33 (4)** (1999), 381-390.

[42] **M. Gendreau, G. Laporte, and R. Séguin,** *"A Tabu Search Heuristic for the Vehicle Routing Problem with Stochastic Demands and Customers"*, Operations Research **44**, 469-477 (1996).

[43] **M. Gendreau, G. Laporte, and R. Séguin,** *"An exact Algorithm for the Vehicle Routing Problem with Stochastic Customers and Demands"*, Transportation Science **29**, 143-155 (1995).

[44] **M. Gendreau, G. Laporte, and R. Séguin,** *"Stochastic Vehicle Routing"*, Eur. J. Oper. Res. **88**, 3-12 (1996).

[45] **M. Gendreau, G. Laporte and F. Semet,** *Solving an Ambulance Location Model by Tabu Search*, Location Science **5** (1997), 75-88.

[46] **M. Gendreau, G. Laporte and F. Semet,** *A Dynamic Model and Parallel Tabu Search Heuristic for Real-Time Ambulance Relocation*, Technical Report CRT-2000-17, Centre de recherche sur les transports, Université de Montréal, Canada (2000).

[47] **B.E. Gillett,** *A Heuristic Algorithm for the Vehicule-Dispatch Problem*, Operations Research **22** (1974), 340-349.

[48] **F. Glover,** *Tabu Search, PartI*,ORSA journal of computing **1** (1989), 190-206.

[49] **F. Glover,** *Tabu Search, PartII*,ORSA journal of computing **2** (1990), 04-32.

[50] **F. Glover and M. Laguna,** *Tabu Search*, Kluwer, 1997.

[51] **W.G. Graves, and B. A. Whinston,** *An Algorithm for the Quadratic Assignment Problem*, Management Science, **17** (1970), 453-471.

[52] **G. Hadley,** *"Nonlinear and Dynamic Programming"*, Addison-Wesley Publishing Co, Inc., Reading, Mass., 1964.

[53] **J. Halpern,** *"The Shortest-Route with Time-Dependent Length of Edges and Limited Delay Possibilities in Nodes"*, Zeitschift fuer Operations Research **21**, 117-124 (1977).

[54] **M.D. Hickman, D.H. Bernstein,** *"Transit Service and Path Choice Models in Stochastic and Time-Dependent Networks"*, Transportation Science **31 (2)**, 129-146 (1997).

[55] **M.D. Hickman and N.H.M. Wilson,** *"Passenger Travel Time and Path Choice Implications of Real-Time Transit Information"*, Transpn. Res.-C **3 (4)**, 211-226 (1995).

[56] **A.V. Hill and W.C. Benton,** *"Modeling Intra-City Time-Dependent Travel Speeds for Vehicle Scheduling Problems"*, J. Opl. Res. Soc. **43 (4)**, 343-351 (1992).

[57] **A. Hill, V. Mabert and D. Montgomory,** *A Decision Support System for the Courrier Vehicule Scheduling Problem*, Omega Int. J. of Mgmt Sci. **16** (1988), 333-345.

[58] **J.N. Hooker and N.R. Natraj,** *Solving a General Routing and Acheduling Problem by Chain Decomposition and Tabu Search*, Transp. Science **29** (1995), 30-44.

[59] **P. Jaillet,** *The Probabilistic Traveling Salesman Problem*, Ph.D. Thesis, Department of Civil Engineering, MIT (1985).

[60] **P. Jaillet,** *"A Priori Solution of a Traveling Salesman Problem in which a Random Subset of the Customers Are Visited"*, Operations Research **36**, 929-936 (1988).

[61] **P. Jaillet,** *Probabilistic Routing Problems in the Plane*, Operational Research **29** (1991), 675-688.

[62] **I.Z. Jarrah and G. Yu,** *A Decision Support Framework for Airline Flight Cancelations and delays*, Transp. Science **27** (1993), 266-280.

[63] **J.J. Jaw, A.R. Odoni, H.N. Psaraftis and N.H.M. Wilson,** *A Heuristic Algorithm for the Multi-Vehicule Many-To-Many Advance Request Dial-A-Ride Problem*, Working Paper MIT-UMTA-82-3, M.I.T., Cambridge Massachusetts (1982).

[64] **J.J. Jaw, A.R. Odoni, H.N. Psaraftis and N.H.M. Wilson,** *A Heuristic Algorithm for the Multi-Vehicule Advance Request Dial-A-Ride Problem with Time Windows*, Transp. Res. **20B** (1986), 243-257.

[65] **A. Jezequel,** *"Probabilistic Vehicle Routing Problems"*, Master's Thesis, Massachusetts Institute of Technology, 1985.

[66] **B. Kalantani et al.,** *An Algorithm for the Traveling Salesman Problem with Pickup and Delevery Customers*, Journal of Operational Research **22** (1985), 377-386.

[67] **A.J. Kleywegt,** *"Dynamic and Stochastic Models with Freight Distribution Applications"*, Ph.D. Dissertation, Massachusetts Institute of Technology, 1996.

[68] **J.G. Klincewicz and B.M. Rosenwein,** *The Airline Exception Scheduling Problem*, Transportation Science **29** (1995), 04-16.

[69] **G. Laporte, F.V. Louveaux and H. Mercure,** *"A Priori Optimization of the Probabilistic Traveling Salesman Problem"*, Operations Research **42**, 543-549 (1994).

[70] **A. Larsen,** *The Dynamic Vehicle Routing Problem*, Ph.D. Dissertation, University of Denmark, 2000.

[71] **S. Lin,** *"Computer Solutions of the Traveling Salesman Problem"*, Bell System Technical Journal **44**, 2245-2269 (1965).

[72] **K. Lund, O.B.G. Madsen, J.M. Rygaard,** *Vehicule Routing Problems with Varying Degrees of Dynamism* , Technical Report IMM-REP-1996-1, Institute of Mathematical Modeling, Technical University of Denmark, Lyngby, Denmark (1996).

[73] **O.B.G. Madsen, H.F. Ravn and J.M. Rygaard,** *A Heuristic algorithm for a Dial-A-Ride Problem with Time Windows, Multiple Capacities and Multiple Objectives*, Annals of Op. Res. **60** (1995), 193-208.

[74] **O.B.G. Madsen, K. Tosti and J. Voelds,** *A Heuristic Method for Dispatching Repair Men*, Annals of Op. Res. **61** (1995), 213-226.

[75] **H.S. Mahmassani, Y. Kim and P. Jaillet,** *Local Optimization Approaches to Solve Dynamic Commercial Fleet Management Problems*, Transportation Research Record **1733** (2000), 71-79.

[76] **C. Malandraki,** *"Time Dependent Vehicle Routing Problems : Formulations, Solution Algorithms and Computations Experiments"*, Ph.D. dissertation, Northwestern University, Evanston, III., 1989.

[77] **C. Malandraki, M.s. Daskin,** *"Time Dependent Vehicle Routing Problems : Formulations, Properties and Heuristic Algorithms"*, Transportation Science **26 (3)**, 185-200 (1992).

[78] **P.H.J. Marguier and A. Cedar,** *"Passenger Waiting Strategies for Overlapping Bus Routes"*, Transportation Science **18 (3)**, 207-230 (1984).

[79] **C.E. Miller, A.W. Tucker and R.A. Zemlin,** *"Integer Programming Formulation on Traveling Salesman Problems"*, J. Assoc. Comput. Mach. **7**, 326-329 (1960).

[80] **H. Min,** *The Multiple Vehicule Routing Problem with Simultaneous Delivery and Pickup points*, Transp. Res. **23A** (1989), 377-386.

[81] **A. Minkoff,** *A Real Time Dispatching of Delevery Vehicules*, Technical Report, Massachusetts Institute of Technology, Cambridge, Mass. (1985).

[82] **A.S. Minkoff,** *A Markov Decision Model and Decomposition Heuristic for Dynamic Vehicule Dispatching* , Operations Research, **41** (1993), 77-90.

[83] **K. Nachtigall,** *"Time-Dependent Shortest-Path Problems with Applications to Railway Networks"*, European Journal of Operational Research **83**, 154-166 (1995).

[84] **A. Orda and R. Rom,** *"Shortest-Path and Minimum-Delay Algorithms in Networks with Time-Dependent Edge-Length"*, J. ACM **37**, 607-625 (1990).

[85] **J.C. Picard, M. Queyranne,** *"The Time-Dependent Traveling Salesman Problem and its Application to the Tardiness Problem in One-Machine Scheduling"*, Operations Research **26 (1)**, 86-110 (1978).

[86] **J.Y. Potvin and J.M. Rousseau,** *Constraint-Direct Search for the Advanced Request Dial-A-Ride Problem with Service Quality Constraints*, in Computer Science and Operations research : New Developpement in the Interfaces, Balci, Sharda and Zenios (eds), pp. 457-474, Pergamon Press (1992).

[87] **W. Powell,** *A Comparative Review of Alternative Algorithms for the Dynamic Vehicule Allocation Problem*, in Vehicule Routing : Methods and Studies, B.L. Golden and A.A. Assad (eds), pp. 249-291, North Holland : Amsterdam (1988).

[88] **W. Powell, Y. Sheffi, K.S. Nickerson, K. Butterbaugh and S. Atherton,** *Maximizing Profits for North American Van Lines Truckload Division : A New Framework for Pricing and Operations*, Interfaces, **18** (1988), 21-41.

[89] **W.B. Powell, P. Jaillet and A. Odoni,** *Stochastic and Dynamic Networks and Routing*, in Network Routing, Handbooks in Operations Research and Management Science, Volume 8, M.O. Ball, T.L. Magnanti, C.L. Monma and G.L. Nemhauser (eds), pp. 141-295, North-Holland : Amsterdam (1995).

[90] **W.B. Powell,** *A Stochastic Formulation of the Dynamic Assignment Problem, with an Application to Truckload Motor Carriers*, Transportation Science **30 (3)** (1996), 195-219.

[91] **W. Powell,** *On Languages for Dynamic Resource Scheduling Problems*, in Fleet Management and Logistics, T.G. Crainic and G. Laporte (eds), pp. 127-157, Kluwer Academic Publishers (1998).

[92] **W.B. Powell, G. Godfrey, K. Papadaki, M. Spivey and H. Topaloglu,** *Adaptive Dynamic Programming for Multistage Stochastic Resource Allocation*, Department of Operations Research and Financial Engineering, Princeton University (2000).

[93] **W.B. Powell, W. Snow and R.K. Cheung,** *Adaptive Labeling Algorithms for the Dynamic Assignment Problem*, Transportation Science **34** (2000), 50-66.

[94] **W.B. Powell, M.T. Towns and A. Marar,** *On the value of Optimal Myopic Solutions for Dynamic Routing and Scheduling Problems in the Presence of User Non Compliance*, Transportation Science **34** (2000), 67-85.

[95] **H.N. Psaraftis,** *A Dynamic Programming Solution to the Single-Vehicule Many-to-Many Immediate Request Dial-A-Riade Problem* , Transportation Science **14** (1980), 130-154.

[96] **H.N. Psaraftis,** *K-Interchange Procedures for Local Search in a Precedence-Constrained Routing Problem*, European Journal of Operational Research **13**

(1983), 391-402.

[97] **H.N. Psaraftis,** *An exact Algorithm for the Single Vehicule Many-To-Many Dial-A-Ride Problem with Time Windows*, Transportation Reaserch **17** (1983), 351-357.

[98] **H.N. Psaraftis,** *Analysis of $O(N^2)$ Heuristic for the Single Vehicule Many-To-Many Euclidian Dial-A-Ride Problem*, Trasportation Reaserch **17B** (1983), 133-145.

[99] **H.N. Psaraftis, J.B. Orlin, D. Bienstock and P.M. Thompson,** *Analysis and solution algorithms of sealift Routing and scheduling problems : Final Report*, Working Paper No. 1700-85, Sloan School of Management, M.I.T. (1985).

[100] **H.N. Psaraftis,** *Scheduling Large Scale Advance-Request Dial-A-Ride Systems*, Amer. J. Math. Management Sc. **06** (1986), 327-367.

[101] **H.N. Psaraftis,** *Dynamic Vehicule Routing Problems*, in Vehicule Routing : Methods and Studies, B.L. Golden and A.A. Assad (eds), pp. 223-249, North Holland : Amsterdam (1988).

[102] **H.N. Psaraftis M.M. Solomon, T.L. Magnanti and T.U. Kim,** *Routing and Scheduling on a Shoreline with release times*, Management Science **36** (1990), 212-223.

[103] **H.N. Psaraftis,** *Dynamic Vehicule Routing : Status and Prospects* , Annals of Operations Research **61** (1995), 143-164.

[104] **H.N. Psaraftis, J.N. Tsitsiklis,** *"Dynamic Shortest Paths in Acyclic Networks with Markovian Arc Costs"*, Operations Research **41 (1)**, 91-101 (1993).

[105] **B. Ran, D.E. Boyce and L.J. Leblanc,** *A New Class of Indtantaneous Dynamic User-Optimal Traffic Assignmrnt Models*, Operations Research, **41** (1993), 192-202.

[106] **H. Rappoport,** *A Planing Heuristic for Military Airlift*, Interfaces **22** (1992), 73-87.

[107] **A.C. Regan, H.S. Mahmassani and P. Jaillet,** *Real-Time Information for Improved Efficiency of Commercial Vehicle Operations*, Proc., First World Congress on Applications of Transport Telematics & Intelligent Vehicle-Highway Systems, **Nov.** (1994).

[108] **A.C. Regan, H.S. Mahmassani and P. Jaillet,** *Improving Efficiency of Commercial Vehicle Operations Using Real-Time Information : Potential Uses and Assignment Strategies*, Transportation Research Record **1493**, 188-198 (1994).

[109] **A.C. Regan, H.S. Mahmassani and P. Jaillet,** *Dynamic Vehicle Allocation for Fleet Management : Operational Changes for Improved Efficiency*, Proc., 2nd World Congress on Applications of Transport Telematics & Intelligent Vehicle-Highway Systems, **Nov.** (1995).

[110] **A.C. Regan, H.S. Mahmassani and P. Jaillet,** *Dynamic Decision Making for Commercial Fleet Operations Using Real-Time Information*, Transportation Research Record **1537**, 91-97 (1995).

[111] **R. Rivard,** *Construction des parcours des véhicules et des Horaires des Chauffeurs pour le Transport des Personnes Handicapées*, publication CRT-240, Centre de recherche sur les transports, Université de Montréal, Canada (1981).

[112] **Y. Rochat and E. Taillard,** *Probabilistic Diversification and intensification in Local Search for Vehicule Routing*, Journal of Heuristics, **1** (1995), 147-167.

[113] **J.M. Rousseau, and S. Roy,** *RAO Répartition Assistée par Ordinateur : la Déscription du Prototype* , publication CRT-564, Centre de recherche sur les transports, Université de Montréal, Canada (1988).

[114] **S. Roy J.M. Rousseau, G. Lapalme and J.A. Ferland,** *Routing and Scheduling for the Transportation of Disables Persons : the Algorithm*, Publication CRT-412, Centre de recherche sur les transports, Université de Montréal, Canada (1985).

[115] **M.W.P. Savelsberg,** *The general Pickup and delevery Problem*, Tranp. Science **29** (1992), 17-29.

[116] **R. Séguin, J.Y. Potvin, M. Gendreau, T.C. Crainic and P. Marcotte** *Real-time Decision Problems : an Operational Research Perspective*, Journal of the Operational Research Society **48** (1997) 162-174.

[117] **R. Rubio Maqueo,** *"Dynamic-Stochastic Vehicle Routing and Inventory Problem"*, Ph.D. Dissertation, Massachusetts Institute of Technology, 1995.

[118] **R. Séguin,** *"Problèmes Stochastiques de Tournées de véhicules"*, Ph.D. Dissertation, Département d'Informatique et de Recherche Opérationnelle, Université de Montréal (1994).

[119] **Y. Shen, J.Y. Potvin, J.M. Rousseau and S. Roy,** *A Computer Assistant for Vehicule Dispatching with Learning Capabilities*, Annals of Operations Reaserch **61** (1995), 189-211.

[120] **M.M. Solomon,** *Algorithms for the Vehicule Routing and Scheduling Problems with Time Window Constraints*, Operations Research **35** (1987), 254-265.

[121] **R.S. Solanki,** *An execution planning Algorithm for Military Airlift*, Interfaces **21** (1991), 121-131.

[122] **D.M. Stein,** *Scheduling Dial-A-Ride Transportation Systems*, Transportation Reaserch **12** (1978), 232-249.

[123] **C. Sutcliffe and J. Board,** *Optimal Solution of a Vehicule Routing Problem : Transporting Mentally Handicapped Adults to an Adult Training Center*, J. Opl Res. Soc. **41** (1990), 61-67.

[124] **M.R. Swihart and J.D. Papastavrou,** *"A Stochastic and Dynamic Model for the Single-Vehicle Pick-up and Delivery Problem"*, Eur. J. Oper. Res. **114**, 447-464 (1999).

[125] **É. Taillard, P. Badeau, M. Gendreau, F. Guertin and J. Y. Potvin,** *A Tabu Search Heuristic for the Vehicle Routing Problem with Soft Time Windows*, Transportation Science **31** (1997) 170-186.

[126] **P.M. Thompson and H.N. Psaraftis,** *Cyclic Transfer Algorithms for Multivehicule Routing and Scheduling Problems*, Operations Research **41** (1993), 935-946.

[127] **P. Trudeau, J.M. Rousseau, J.A. Ferland and J. Choquette,** *An Operations Research Approch for the Planning and Operating of an Ambulance Service*, INFOR **27** (1989), 95-113.

[128] **J. Yang, P. Jaillet and H.S. Mahmassani,** *On-line Algorithms for Truck Fleet Assignment and Scheduling under Real-Time Information*, Transportation Research Record **1667** (1999), 107-113.

[129] **W.H. Yang, K. Mathur and R.H. Ballou,** *"Stochastic Vehicle Routing Problem with Restocking"*, Transportation Science **34 (1)**, 99-112 (2000).

[130] **C.D.J. Waters,** *"Vehicle Routing Problems with Uncertainty and Omitted Customers"*, Journal of the Operational Research Society **40**, 1099-1108 (1989).

[131] **I. Webb,** *Period and Phase of Customer Replenishment : A New Approch to Inventory/Routing Problems*, Technical Report, Massachusetts Institute of Technology, Cambridge, Mass. (1989).

[132] **N.H.M. Wilson, J.M. Sussman, H.K. Wang and B.T. Higonnet,** *Scheduling Algorithms for Dial-a-Ride Systems*, Report USL TR-70-13, Urban Systems Laboratory. M.I.T. (1971).

[133] **N.H.M. Wilson and N.H. Colvin,** *Computer Control of the Rochester Dial-a-Ride System*, Report R77-31, Dept. of Civil Engineering. M.I.T. (1977).

[134] **A.K. Ziliaskopoulos,** *"Optimum Path Algorithms on Multidimensional Networks : Analysis, Design, Implementation and Computational Experience"*, Ph.D. Dissertation, University of Texas at Austin, 1994.

[135] **A.K. Ziliaskopoulos and H.S. Mahmassani,** *"Time-Dependent, Shortest-Path Algorithm for Real-Time Intelligent Vehicle Highway System Applications"*, Transportation Research Record **1408**, 94-100 (1993).